UNIVERSITI TEKNOLOGI MALAYSIA

DECLARATION OF THESIS / UNDERGRADUATE PROJECT REPORT AND COPYRIGHT

Author's full name : WOO CHEE YOONG

Date of Birth : November 14TH 1990

Title : HARDWARE DESIGN ON TOUCH SCREEN

Academic Session : **2013/2014**

I declare that this thesis is classified as:

☐	**CONFIDENTIAL**	(Contains confidential information under the Official Secret Act 1972)*
☐	**RESTRICTED**	(Contains restricted information as specified by the organization where research was done)*
☑	**OPEN ACCESS**	I agree that my thesis to be published as online open access (full text)

I acknowledged that Universiti Teknologi Malaysia reserves the right as follows:

1. The thesis is the property of Universiti Teknologi Malaysia
2. The Library of Universiti Teknologi Malaysia has the right to make copies for the purpose of research only.
3. The Library has the right to make copies of the thesis for academic exchange.

Certified by:

SIGNATURE	**SIGNATURE OF SUPERVISOR**
901114-14-6779	DR. RABIA BAKHTERI
(NEW IC NO/PASSPORT)	**NAME OF SUPERVISOR**
Date: 22th JUNE 2014	Date: 22th JUNE 2014

NOTES: * If the thesis is CONFIDENTAL or RESTRICTED, please attach with the letter from the organization with period and reasons for confidentiality or restriction.

HARDWARE DESIGN ON TOUCH SCREEN

WOO CHEE YOONG

UNIVERSITI TEKNOLOGI MALAYSIA

"I hereby declare that I have read this report and in my

opinion this thesis is sufficient in terms of scope and quality for the
award of the degree of Bachelor of Engineering (Electrical - Microelectronic)"

Signature	:	
Name	:	DR.RABIA BAKHTERI
Date	:	June 19, 2014

HARDWARE DESIGN ON TOUCH SCREEN

WOO CHEE YOONG

A report submitted in fulfilment of the
requirements for the award of the degree of
Bachelor of Engineering (Electrical - Microelectronic)

Faculty of Electrical Engineering
Universiti Teknologi Malaysia

JUNE 2014

I declare that this report entitled *"HARDWARE DESIGN ON TOUCH SCREEN"* is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature	:	
Name	:	WOO CHEE YOONG
Date	:	June 19, 2014

Special thanks to my beloved family members who always there for me, my friends and seniors who asisting, accompanying me now and also to my supervisor who guide me through the research's hardships.

ACKNOWLEDGEMENT

I would like to express my appreciation and thankfulness to the people who directly or indirect contributes and helps me throughout the entire process in carrying out this project. This project will not be completed with the helps, supports, assistance and precious advices from every one of you as mentioned here.

First and foremost, I would like to express my greatest gratitude to my final year project supervisor, Dr Rabia Bakhteri for giving me this opportunity to handle such challenging project. She supervises and provides me guidance throughout the entire project research process, and she also point out my personal weakness so that I can improve myself for future challenges.

In addition, I would like to express my thankfulness to my families and friends around me that give me strength and mental support to continue this project. Nevertheless, I would like to deliver my sincere appreciation to Mr Chan Mun Kit as part of team in DreamCather innovation competition and assist me throughout the project.

Lastly, I would like to deliver my sincere gratitude to everyone who is involved in this project.

ABSTRACT

This Project is conducted using Altera DE2 development board and Terasic touch screen. The main purpose of this project is to create a digit recognition device. This project is divided into two parts. One is to build a Convolutional Neural Network block which is able to recognize digit, another part is to integrate the TRDB-LTM touch panel and Field Programmable Gate This project is focuses on painting application using finger on a touch screen and save the image data file. There is no software approach and done using a FPGA chip with Verilog code. In this project, there is no proper device driver to connect the touch panel and the FPGA board to perform drawing application. Then, another problem is that there is no graphical user interface (GUI) for the touch panel. This GUI should also display the output image file and other text. Furthermore, most of the character recognition systems are software based. The performance if the system can be improved if the hardware design is implemented in this system. Hence, this project consists of three main objectives, first objective is to build a suitable device driver which enables the touch panel to communicate properly with the Altera DE2 board. After that, design a GUI on the touch sceen which allows the user to draw digits, gives different instructions by touching different coordinates and dislays text. Then, build a pure hardware based system.

ABSTRAK

Projek ini dijalankan dengan menggunakan "Altera DE2 board" dan Terasic skrin sentuh. Tujuan utama projek ini adalah untuk mewujudkan peranti pengiktirafan digit. Projek ini dibahagikan kepada dua bahagian. Satu adalah untuk membina satu blok convolutional Neural Network yang dapat mengenali angka, bahagian lain adalah untuk mengintegrasikan panel sentuh TRDB-LTM dan "Field Programmable Gate Array". Projek ini memberi tumpuan terhadap aplikasi lukisan menggunakan jari pada skrin sentuh dan menyimpan fail data imej . Tidak ada pendekatan perisian dan dilakukan dengan menggunakan cip FPGA dengan kod Verilog. Dalam projek ini, tidak ada pemacu peranti yang betul untuk menyambung panel sentuh dan papan FPGA untuk melaksanakan permohonan lukisan. Kemudian, satu lagi masalah adalah bahawa tidak ada antara muka pengguna grafik (GUI) untuk panel sentuh. GUI ini juga perlu memaparkan fail imej output dan teks yang lain. Tambahan pula, sebahagian besar sistem pengiktirafan watak adalah perisian berasaskan. Prestasi jika sistem boleh diperbaiki jika reka bentuk perkakasan dilaksanakan dalam sistem ini. Oleh itu, projek ini terdiri daripada tiga objektif utama, Objektif pertama adalah untuk membina sebuah pemacu peranti yang sesuai yang membolehkan panel sentuh untuk berkomunikasi dengan betul dengan "Altera DE2 board". Selepas itu, mereka bentuk GUI pada sceen sentuhan yang membolehkan pengguna untuk menarik digit, memberikan arahan yang berbeza dengan menyentuh koordinat yang berbeza dan dislays teks. Kemudian, membina sistem perkakasan.

TABLE OF CONTENTS

CHAPTER	TITLE		PAGE
	DECLARATION		ii
	DEDICATION		iii
	ACKNOWLEDGEMENT		iv
	ABSTRACT		v
	ABSTRAK		vi
	TABLE OF CONTENTS		vii
	LIST OF TABLES		ix
	LIST OF FIGURES		x
	LIST OF APPENDICES		xii
1	**INTRODUCTION**		**1**
	1.1	Background of the Project	1
	1.2	Problem Statement	2
	1.3	Objective	3
	1.4	Project Scope	4
	1.5	Project Overview	4
	1.6	Thesis Structure	5
2	**LITERATURE REVIEW**		**6**
	2.1	Types of User Interfaces	6
		2.1.1 The Command Line Interface	6
		2.1.2 The Menu Driven Interface	7
		2.1.3 The Graphical User Interface	8
	2.2	Touch Screen Interface	8
	2.3	Related Works	9
		2.3.1 Digital Photo Frame	9
		2.3.2 Galaxian Game on Altera DE2-115 FPGA Architecture	11
		2.3.3 Touch Pong	14

		2.3.4 Painter Demonstration	16
	2.4	LTM User Manual	18
3	**RESEARCH METHODOLOGY**		**20**
	3.1	Process flow	21
	3.2	Touch Screen Controller	21
	3.3	The Serial Interface of the Analog Digital (AD) Converter	23
	3.4	Drawing Application	24
	3.5	Memory for Image Storage	26
	3.6	Erosion Function	27
4	**RESULT AND DISCUSSION**		**29**
	4.1	LCD module	29
	4.2	Memory Initialization File	32
	4.3	Touch Screen module	33
	4.4	Drawing Application	34
	4.5	Image Erosion	39
5	**PROJECT MANAGEMENT**		**40**
	5.1	Project Schedule	40
6	**CONCLUSION**		**42**
REFERENCES			**44**
Appendix A			46

LIST OF TABLES

TABLE NO.	TITLE	PAGE
2.1	General specification of the touch screen	19
4.1	Signal Assignments for the LCD Module.	29
4.2	Corresponded character codes and character patterns.	31
4.3	Memory Address Assignments	32
4.4	Inputs Digit and Outputs	38
5.1	Gantt Chart Semester 1	40
5.2	Gantt Chart Semester 2	41

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
1.1	Process of Digit Recognition	2
1.2	Project Overview	5
2.1	Command Line Interface	7
2.2	Menu Driven Interface	7
2.3	Graphical User Interface	8
2.4	Resistice Touch Screen.	9
2.5	Capacitive Touch Screen.	9
2.6	Block Diagram of the Digital Photo Frame System	10
2.7	Result of Digital Photo Frame	11
2.8	Hardware Blocks	12
2.9	Reconfigurable IP Core Operation Mode	12
2.10	Start Interface	13
2.11	Game Running Interface	13
2.12	Touch Screen Processor	15
2.13	Block Diagram of the Photo Demonstration	15
2.14	Block Diagram of the Painter Demonstration	17
2.15	Painter Demostration	17
2.16	The TRDB-LTM Package	18
2.17	The block Diagram of the LTM	19
3.1	System Architecture	20
3.2	Process Flow	21
3.3	Serial interface of the LCD Touch Panel Module and AD7843	22
3.4	Conversion Timing of the Serial Port Interface	23
3.5	Block Diagram of the Drawing Application.	25
3.6	Square Dot with Black Colour Creation	26
3.7	Mif fle in Quartus II	26
3.8	36x27 Pixels Image Data File.	27
3.9	32x32 Pixels Image Data File.	28
4.1	Schematic Diagram of the LCD Module	30
4.2	Result of LCD Module.	32

4.3	Vecor Waveform of Mif file.	33
4.4	TRDB-LTM Touch Screen and DE2-35 Board.	34
4.5	480x1440 Bitmap Image	35
4.6	Compilation Report	36
4.7	Digit Recognition System on Touch Screen.	36
4.8	Image Erosion.	39

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	Drawing Application	46

CHAPTER 1

INTRODUCTION

1.1 Background of the Project

With recent technology enhancement, more and more activities are being computerized. Nowadays, many tasks and transactions are done online especially financial system. Many banking options are available like automated teller machine (ATM) and online shopping. Currently, these system use password as user authentication to prove identity or access approval which is not secure, can be forget and stolen.

In fact, the best security measures are those that use user biometrics which identify the user by recognized the user signature, face or hand. An example of secure online banking is the e-cheque system. This is a proposed system which consists of an embedded e-cheque device which can be used by user to perform secure transactions online. Signature is proposed to verify user identity and such a system requires the use of a touch screen to capture text written by user and an intelligent system like Convolutional Neural Network (CNN) to recognize the text.

CNN is a special type of neural network that is widely used in pattern recognition such as face, speech and handwriting recognition. This computational model is inspired by theanimal's central nervous systems and they are usually presented as systems of interconnected "neurons" that can compute values from inputs by feeding information through the network. Neural networks have been successfully applied to chemical process control, aerospace application, continuous production of high-quality parts and robot arm control. However, this report is focused on the integration of the touch panel and FPGA board. The first prototype of signature recognition e-cheque system would be the hardware design of a digit recognition system.

There are some securities concerns, even though online banking sites are heavily encrypted, with the developing technology, it's hard to rule out the "hackers" who may access your bank accounts. Software based systems are more susceptible to brute force attacks and other method of techniques. For example, online check banking is not very secure due to the scanner use image processing technique to identify the figure or number on the cheque which is more to software based and it can be easily alter or modified by modify its program.

On the other hand, hardware-based system can increase performance by off-loading encryption from the host system and it does not require any type of driver installation or software installation on the host PC, protects against the most common attacks, such as cold boot attacks, malicious code and brute force attacks. In addition, most of the neural networks are run as software general processing unit, by implementing neural networks in hardware, the process can be off-loaded from the host processor. Hence, the speed of the system can be enhanced by using hardware approach. So basically, our objective in this project is to create more secure and faster pattern recognition system by using hardware approach and implement it on FPGA and the scope of this project is limited to creating a hardware device driver for the touch screen amd a Graphical User Interface (GUI) for the E-cheque application.

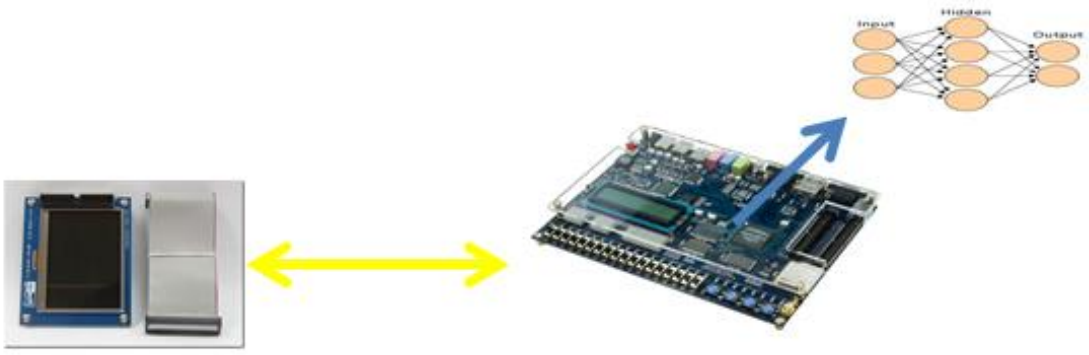


Figure 1.1 Process of Digit Recognition

1.2 Problem Statement

In this project, there is no proper device driver to interface between the touch panel and the FPGA board to perform drawing application. A device driver is a small program that enables a computer and a device to communicate with each

other. Nowadays, computer operating systems usually come with 'pre-installed' device drivers for the current models of the popular devices. Also, some older devices usually come on a disk with the device driver or can be downloaded from the manufacturer's website.

Then, another problem is that there is no graphical user interface (GUI) for the touch panel. GUI is a program using graphical interface that takes advantage of the computer's graphics capabilities to make the program easier to use. Well-designed graphical user interfaces can free the user from learning complex command languages. In this case, the system needs a GUI for the user to enter input such as hand drawn digits and issues commands such as cancel and recognize. This GUI should also display the output image file and other text.

Furthermore, most of the character recognition systems are software based. However, the performance of the system can be improved if hardware design is implemented in this system. This is because software approach is susceptible to viruses due to harmful software on the computer system can influence the proper operation. Besides, the speed of the software approach can be influenced by the other applications which are running on the computer system, the more applications are running on the computer system, the software system is slower. Moreover, the data in the software is corrupted from any power loss while writing data.

1.3 Objective

This project consists of three main objectives:

1. To build a suitable device driver which enables the TRDB-LTM touch panel to communicate properly with the Altera DE2 board.
2. To design a GUI on the touch screen which allows the user to draw digits, gives different instructions on desired touch point coordinates and displays proper image file result and text.
3. To build a pure hardware implementation for the system with high speed response, high security and stand alone device.

1.4 Project Scope

- DE2 Development and Education Board

The FPGA board is used as the hardware platform for implementation of this project.

- TRDB_LTM (LTM) Touch Panel

The touch panel is a touch screen with 40 GPIO pins that can be connected directly to the Altera DE2 FPGA board.

- Altera Quartus II

The software is a platform that can be used to design hardware block for Altera DE2 board, design device driver for the touch screen and perform drawing application.

- Verilog Code

The code is hardware description language used in Altera Quartus II that can design the functionality of the touch screen and its implementation.

- No neural network

Neural network design is beyond the scope of this project.

1.5 Project Overview

The e-cheque application is a project implemented on Altera DE2 development and education board. This application uses the touch screen as the input/output interface and a convolutional Neural Network that is used to recognize hand drawn digits.The main purpose of this project is to create a digit recognition device.

This project is divided into two parts. One is to build a Convolutional Neural Network block which is able to recognize digit, another part is to integrate the TRDB-LTM touch panel and Field Programmable Gate arrays (FPGA) board.

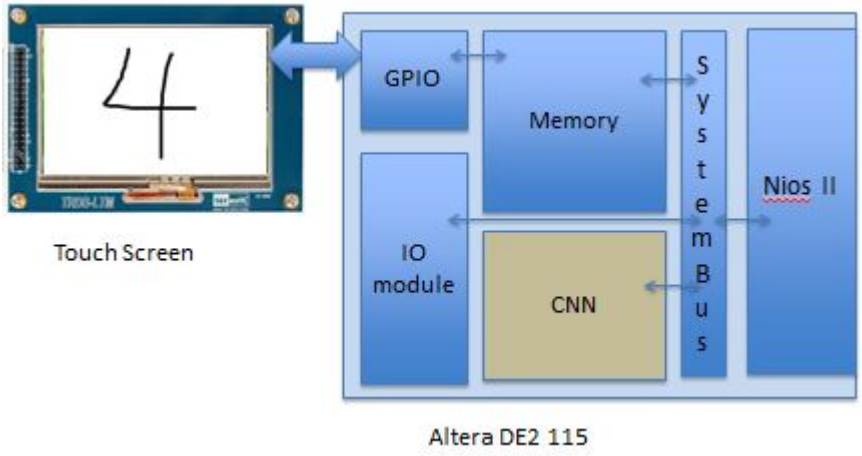


Figure 1.2 Project Overview

1.6 Thesis Structure

This thesis consists of six chapters. Chapter one has discussed the background of the project, problem statement, objective, project scope and project overview. Chapter two describes the specification, theory, information as well as the previous works done on the touch screen. For chapter three, research methodology used in this thesis is explained including the process flow, system architecture and algorithms used for the drawing application and image processing. Chapter four shows the experimental results and followed by analysis and discussion. Chapter five shows the project management. Last but not least, conclusion and future works will be discussed in chapter six.

CHAPTER 2

LITERATURE REVIEW

This Chapter introduces that the theoretical background for this project. Some literature review has been done on some previous research works and product of TRDB-LTM touch panel and DE2 board. Some of the components and designs are being considered when designing the project.

2.1 Types of User Interfaces

The platform for communication between human and computer is called user interface. With a suitable user interface, the user is able to control the machine effectively in a more simple way. Besides, user interface comprised of hardware and software design to show the expected outputs and to capture the inputs to control the computer. There are three main types of user interface and they are described on the next sections.

2.1.1 The Command Line Interface

The comand line interface is a user interface that manipulates the computer system by typing command as input and gets the corresponding output from the monitor. The strengths of this interface is that is is simple and fast for the expert user who is able to memorize and implement the commands and this interface uses less memory. However, command line interface is seldom used by normal users because there are a lot of commands and it is difficult to memorize. Figure 2.1 displays example of command inputs and the output obtained.

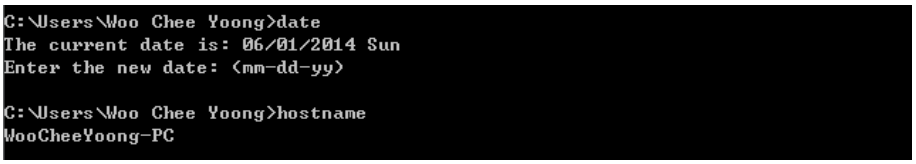


Figure 2.1 Command Line Interface

2.1.2 The Menu Driven Interface

This user interface allows the user to manipulate the computer or device by choosing options from screens or menus. Menu driven interface is widely used in electronic devices like smart phone and automated teller machine (ATM) which are presented with menus and the user can select options from the menu. The strengths of menu driven interface are user friendly due to the user not having to memorize many commands. Furthermore, this interface is easy to understand by user due to guidelines from the menu. However, this interface can be time consuming due to multiple screen menus that need to be navigated and finding the location of a specific command.

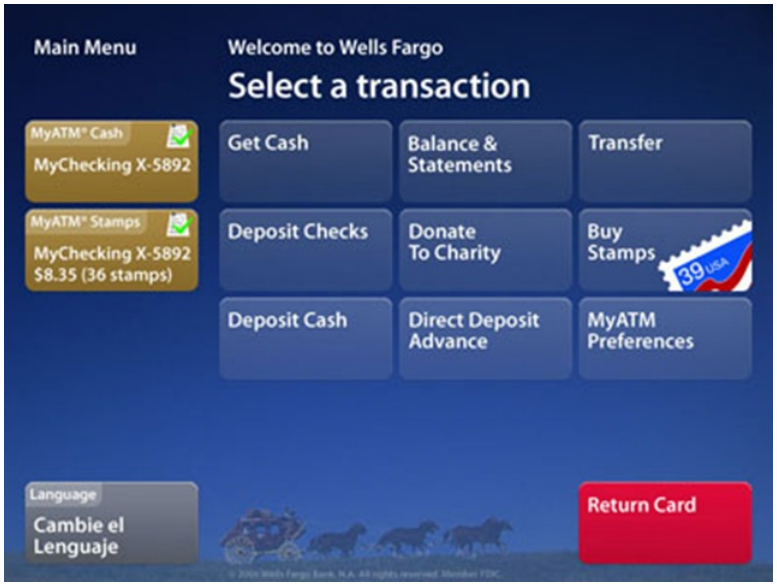


Figure 2.2 Menu Driven Interface

2.1.3 The Graphical User Interface

User interface that allows user to manipulate the computer by taking advantage of the computer's graphics to achieve more user friendly operation. GUI based operating system such as Windows and the Mac OS are very popular for normal users due to their simplicity and case of use. The strengths are user friendly and easy to learn since the users do not have to memorize plenty of commands. Furthermore, GUI is especially suitable for some applications like selecting text or drawing pictures. Besides that, users can run many programs at the same time in different windows.

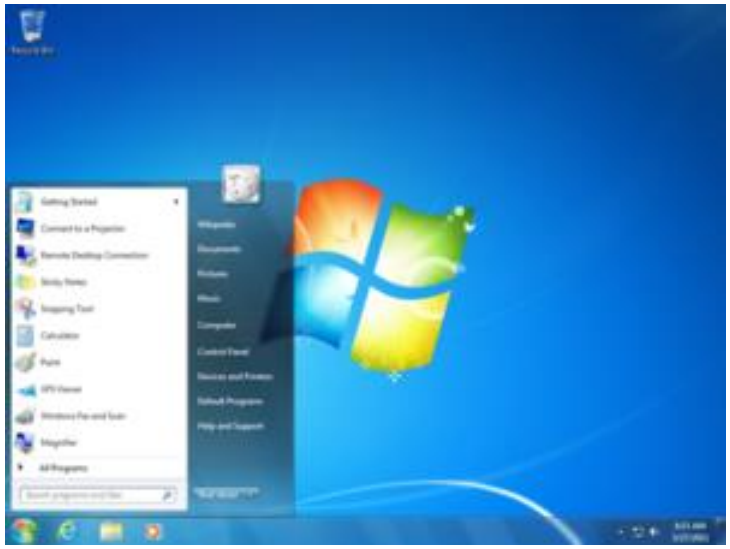


Figure 2.3 Graphical User Interface

2.2 Touch Screen Interface

There are two main types of touch screens which are resistive and capacitive touch screens. This section discusses the differences of these two types of touch screens. For resistive touch screen, when the screen is touched the plastic deforms so that the two ITO (a thin film of conductive material usually Indium Tin Oxide) films meet, and by measuring the resistance of both layers at their point of contact it's possible to get an accurate measurement of the touch position. Whereas for capacitive, as your finger comes close to a capacitor it changes the local electrostatic field, and the system constantly monitors each tiny capacitor to discover exactly where the finger touched the screen: because the measurement points are discrete, it's possible to tell whether several fingers are all touching the screen at once, unlike with a resistive unit.

However, such a tiny capacitance is difficult to measure and requires complex filtering to eliminate noise.

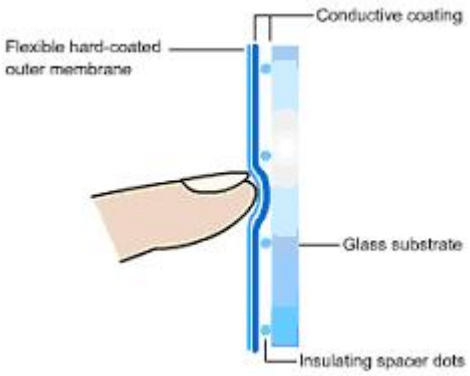


Figure 2.4 Resistice Touch Screen.

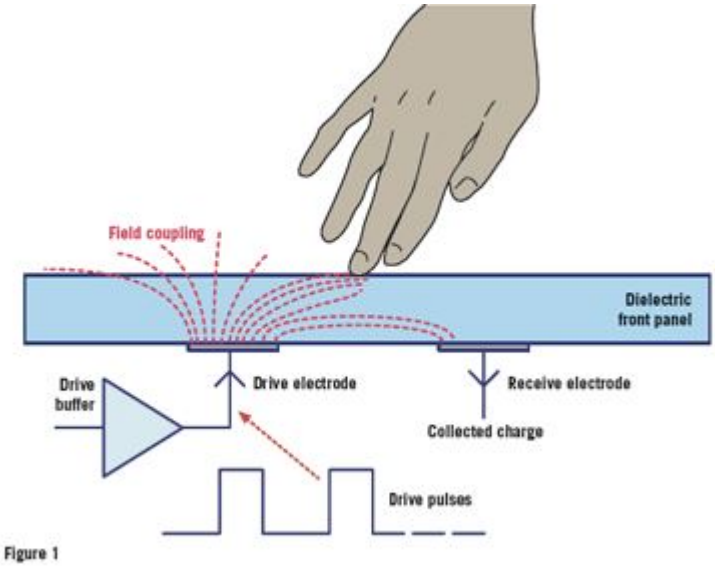


Figure 2.5 Capacitive Touch Screen.

2.3 Related Works

2.3.1 Digital Photo Frame

The work described in [14] is the design of a digital photo frame, using an FPGA board, a VGA monitor, the on board push buttons and SD card reader-module

plugged in to the FPGA board. The final product could read a specially formatted image file from an SD card, put it in the SDRAM on the FPGA board and from there transfer it row by row to a BRAM row-buffer. The VGA controller then read the rows from the BRAM and showed them on the VGA monitor.

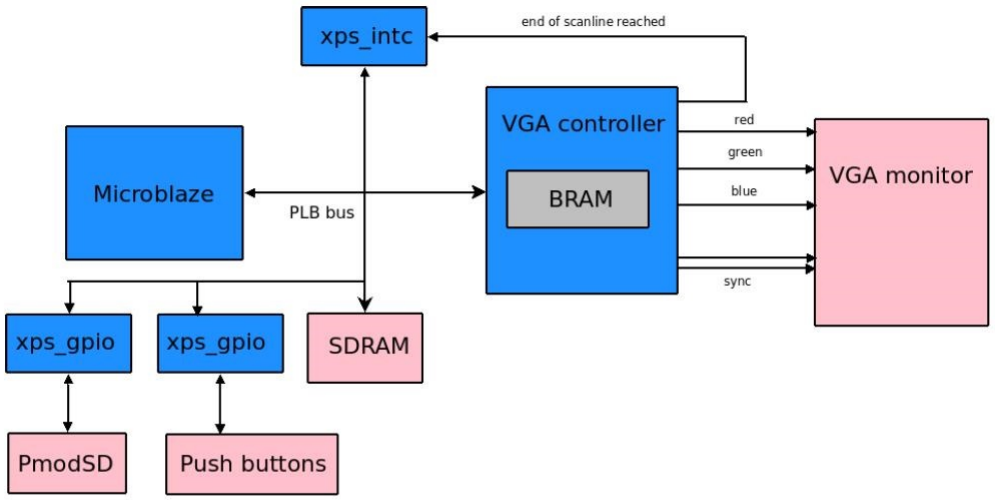
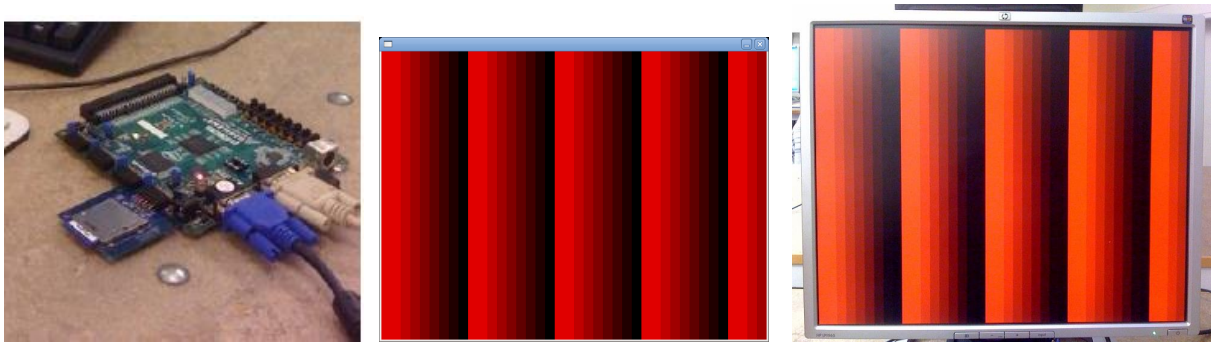


Figure 2.6 Block Diagram of the Digital Photo Frame System[14]

Figure 2.7 shows the system consists of the following hardware components

- Digilent Nexys 2 1200 FPGA board with a Xilinx Spartan 3E FPGA with 16MB of Micron SDRAM & 16MB of Intel StrataFlash ROM
- 4 on board push buttons
- PmodSD - SD card reader module for the FPGA board
- VGA controller with on board BRAM
- Interrupt controller
- VGA monitor



(a) Implementation Board. **(b)** 'Red Gradients' Output **(c)** VGA Monitor Output

Figure 2.7 Result of Digital Photo Frame[14]

2.3.2 Galaxian Game on Altera DE2-115 FPGA Architecture

The work in [21] described on embedded systems based on general-purpose software development model and development process, including the design of hardware platform and software. The design of the hardware platform was based on the Altera's DE2-115 Series development board, the software platform was Nios II 10.1 and the programming language is C and VHDL hardware description language. Based on the SOPC tool, they designed the reconfigurable IP cores of the VGA display, Terasic LTM touch screen. The game was controlled not only by the keyboard, but also through touch screen directly. With Galaxian game as an example, they designed an embedded game based on GUI. The results show that the game system is human-computer interaction friendly and it has quick response and action.

LTM touchscreen driver mainly relies to the liquid-crystal display (LCD) touch screen interrupt input pin position and data to determine coordinates of the touch action. The serial peripheral interface (SPI) bus driver judge interrupt pin level, and then locates the input pin by shifting the position coordinates and stores in the buffer register finally. That is transferred to the application layer of the software by the Avalon bus; the software reads the coordinates so as to control the game.

The system is divided into hardware and software components. The hardware provides control interface and perform tasks operations for the peripherals. The hardware structure is shown in Figure 2.6.

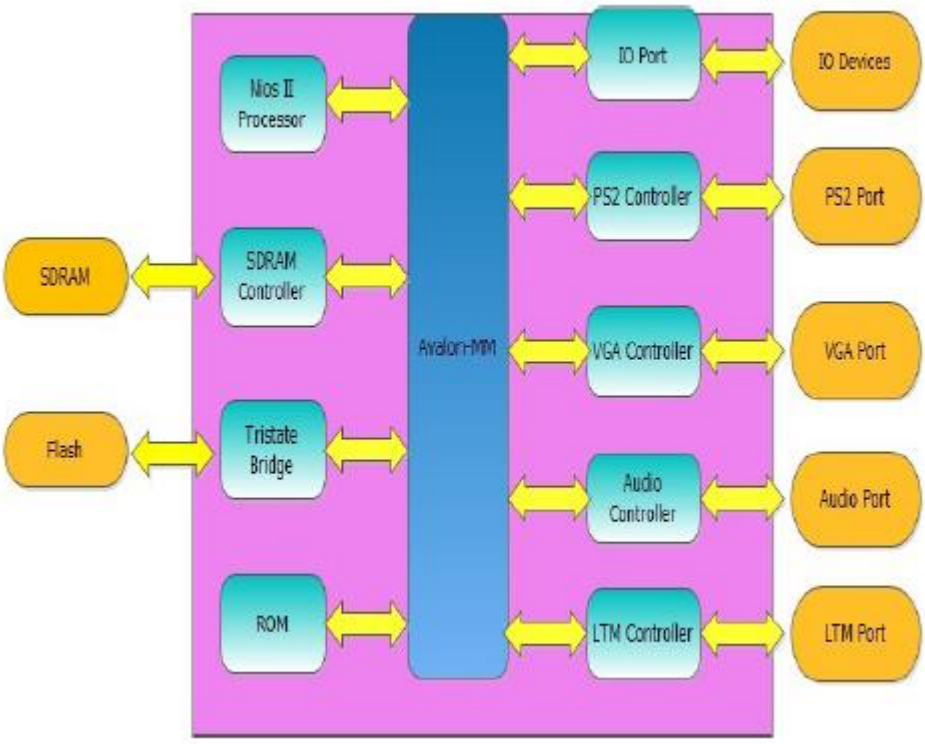


Figure 2.8 Hardware Blocks [21]

So we can use the software to configure some important parameters to control the operation of the IP module. Operate mode is shown in Figure 2.9.

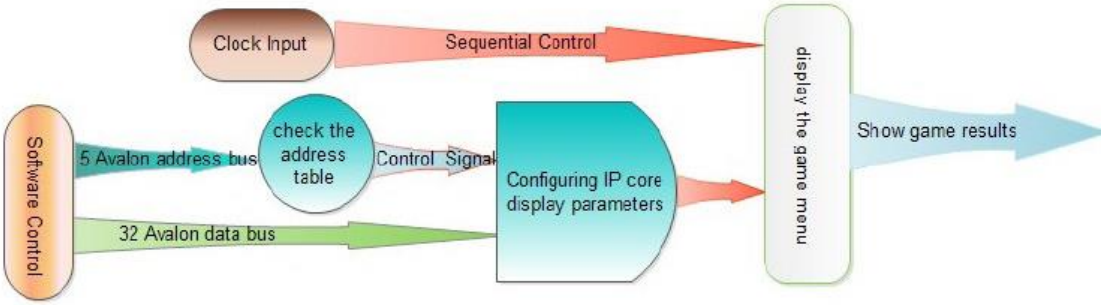


Figure 2.9 Reconfigurable IP Core Operation Mode [21]

The Quartus II software is used to compile the hardware design blocks and generate the final .sof file; download the file to DE2-115 board, and then Nios II Integrated development environment (IDE) can be used to compile the software blocks

needed to run the game. Curing process is done by Nios II IDE Tool on the toolbar under the Flash Programmer to achieve. After curing, it can realize the power of hardware and software automatically loads the program, in order to achieve the game offline. Game runs effect diagram is following.

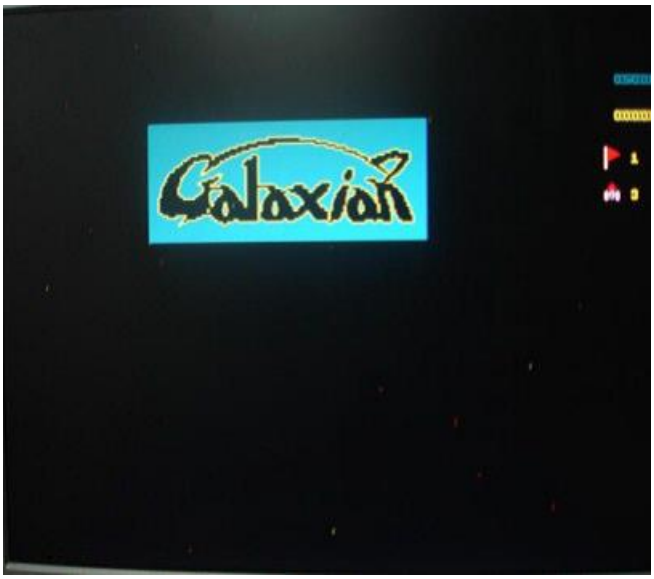


Figure 2.10 Start Interface [21]

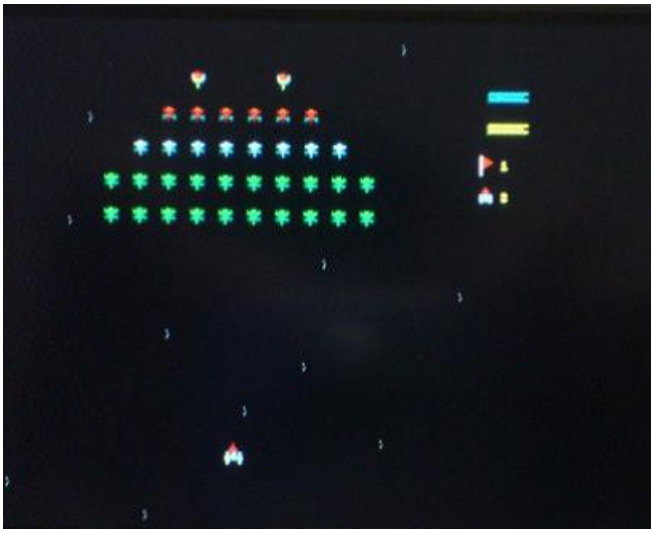


Figure 2.11 Game Running Interface [21]

2.3.3 Touch Pong

This project [22] is conducted using the Altera DE2 development board to implement a touch-screen ping pong game. It was a player versus player game with a specified rule. Player serves and receives the ball by touching the screen connected to DE2 board. To do so, set up the interface between touch screen and DE2 board. The Ping Pong game is an extension of the real Ping Pong game. Set a few new rules for the game. For example, the ball can bounce on the two horizontal sides of the screen and once the ball hits the perpendicular side of the screen, the game is over. To implement the Ping pong game, the project will involve both hardware set up and software programming. Especially, due to the control of the touch screen, the hardware set up will take the most of the work.

• For the hardware part, the major workload is to set up the touch screen and interface. Moreover, the display of the game graphics can also take some efforts.

• For the software part, the difficulty lies in how we realize the algorithm of the Ping pong Game. What's more, add interruption to transmit the coordinate of racket and ball.

In this project, there are two major hardware devices: FPGA board and LTM touch screen. Incorporate VGA display with the TRDB_LTM Kit to develop the application using a digital touch panel on an Altera DE2 board.

1. VHDL(compiled with Quartus 7.2 and Nios II) will be used for the interconnections of hardware.
2. C will be also employed to handle the hardware implementation.
3. The Terasic LCD Touch Panel Module(LTM) board is a displayer and a controller.
4. A 40-pin IDE cable will be used for connecting between the LTM and the DE2 board.

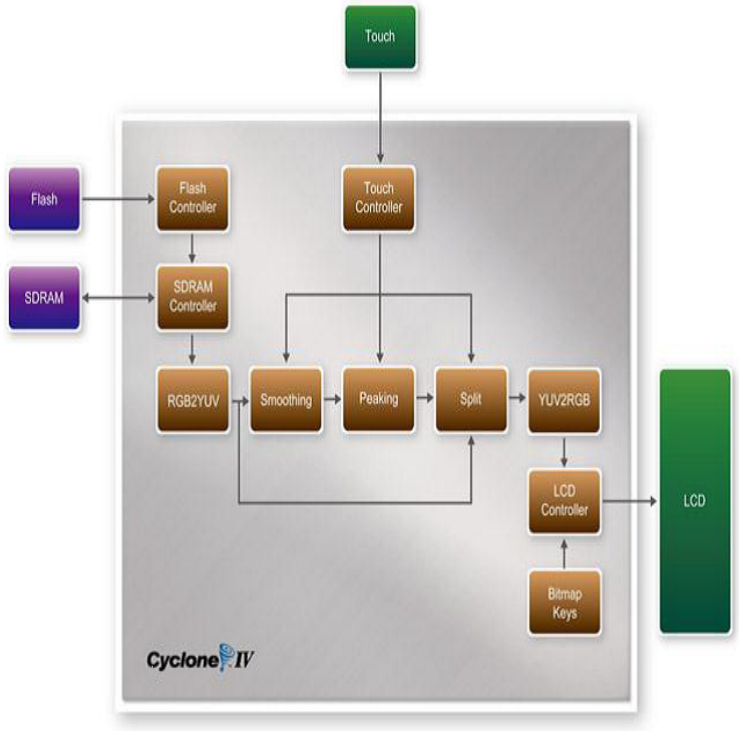


Figure 2.12 Touch Screen Processor[22]

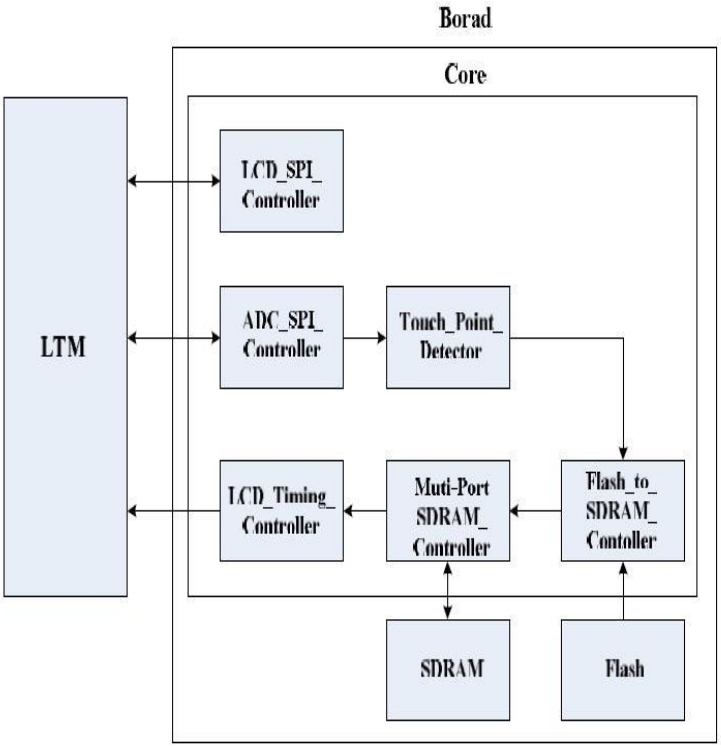


Figure 2.13 Block Diagram of the Photo Demonstration[22]

As soon as the bit stream is downloaded into the FPGA, the register values of the LCD driver IC using to control the LCD display function will be configured by the LCD_SPI_Controller block, which uses the serial port interface to communicate with the LCD driver IC. Meanwhile, the Flash_to_SDRAM_Controller block will read the RGB data of one picture stored in the Flash, and then write the data into SDRAM buffer. Accordingly, both the synchronous control signals andthe picture data stored in the SDRAM buffer will be sent to the LTM via the LCD_Timing_Controller block. When users touch LTM screens, the x and y coordinates of the touch point will be obtained by the ADC_SPI_Controller block through the ADC serial port interface. Then the Touch_Point_Detector block will determine whether these coordinates are in a specific range. If the coordinates fit the range, the Touch_Point_Detector block will control theFlash_to_SDRAM_Controller block to read the next or previous picture's data from the Flash and repeat the steps as mentioned before to command the LTM to display the next or previous picture.

2.3.4 Painter Demonstration

The tutorial found in [3] shows how to manipulate LCD and touch controller to build a paint application based on SOPC Builder and Altera VIP Suite. The multi-touch gestures and single-touch coordinates operation is shown on this demonstration.

For multi-touch processing, a Terasic Memory-Mapped IP is used to retrieve the user input, including multi-touch gesture and single-touch coordinates.

The hardware system block diagram of this demonstration is shown on Figure 2.14. For LCD display processing, Altera Video and Image Processing Suite (VIP) as basic to develop the reference design. Display content is read from the associated video memory by using the Frame Reader VIP. Then, VIP Video Out is used to display the display content. The display content is filled by NIOS II processor according to users' input.

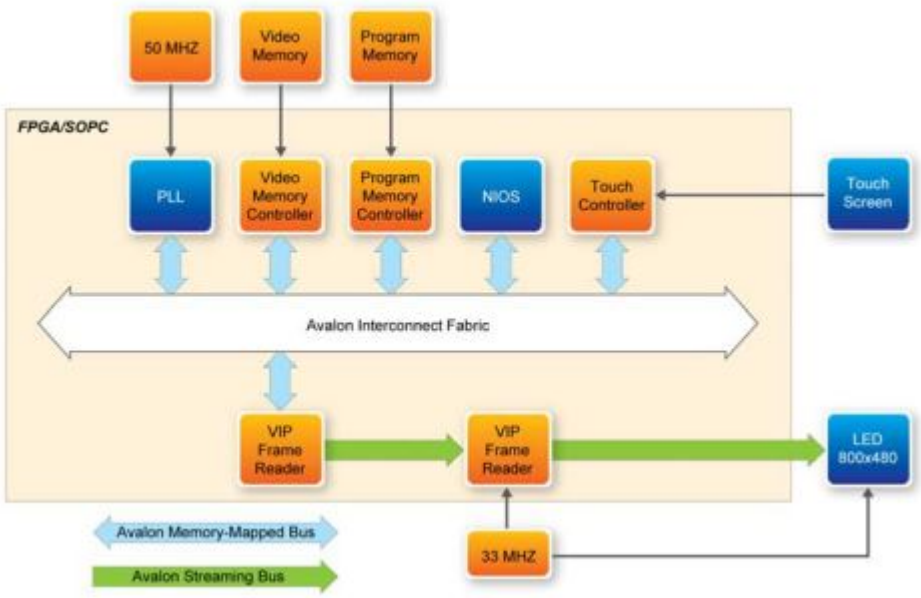
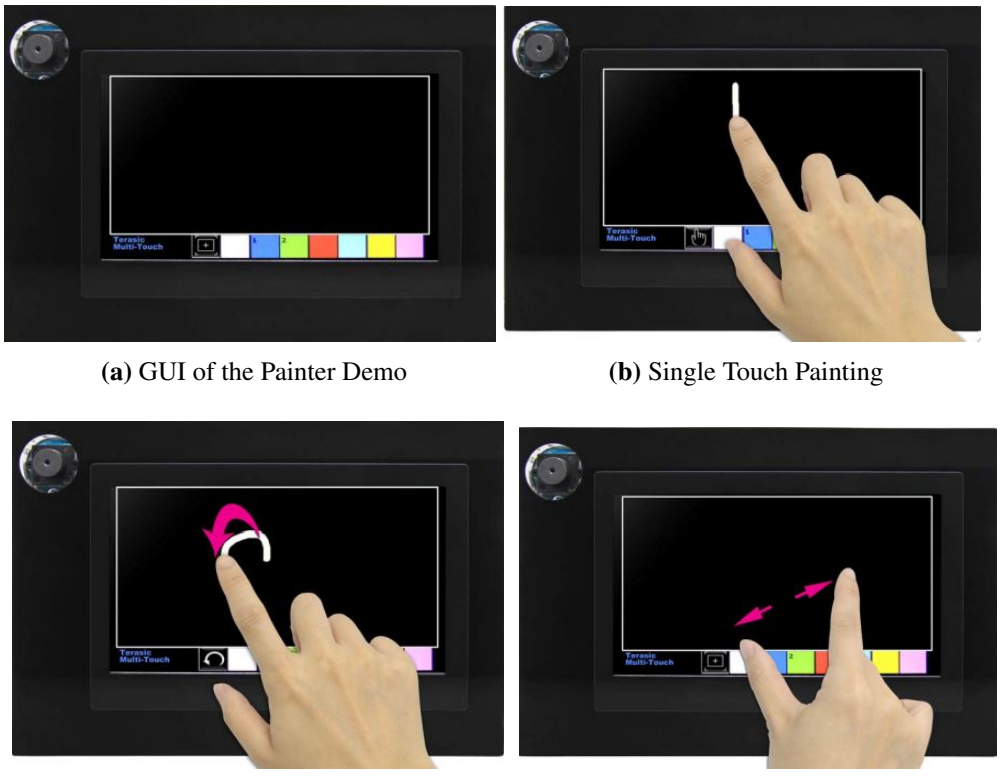


Figure 2.14 Block Diagram of the Painter Demonstration.[3]



(a) GUI of the Painter Demo

(b) Single Touch Painting

(c) Counter-clockwise Rotation Gesture

(d) Zoom-in Gesture

Figure 2.15 Painter Demostration[3]

2.4 LTM User Manual

The LTM consists of three major components: LCD touch panel module, AD converter, and 40-pin expansion header. All of the interfaces on the LTM are connected to Altera DE4/DE2-115/DE2/DE1 board via the 40-pin expansion connector. The LCD and touch panel module will take the control signals provided directly from FPGA as input and display images on the LCD panel. Finally, the AD converter will convert the coordinates of the touch point to its corresponding digital data and output to the FPGA via the expansion header.

Figure 2.16 The TRDB-LTM Package.[2]

The feature set of the LTM is listed below:

1. Equipped with Toppoly TD043MTEA1 active matrix color TFT LCD module.
2. Support 24-bit parallel RGB interface.
3. 3-wire register control for display and function selection.
4. Built-in contrast, brightness, and gamma modulation.

5. Converting the X/Y coordination of the touch point to its corresponding digital data via the Analog Devices AD7843 AD converter.

Table 2.1: General specification of the touch screen[2]

Item	Description	Unit
Display Size (Diagonal)	4.3	Inch
Aspect ratio	15:9	-
Display Type	Transmissive	-
Active Area (HxV)	93.6 x 56.16	mm
Number of Dots (HxV)	800 x RGB x480	dot
Dot Pitch (HxV)	0.039 x 0.117	mm
Color Arrangement	Stripe	-
Color Numbers	16Million	-

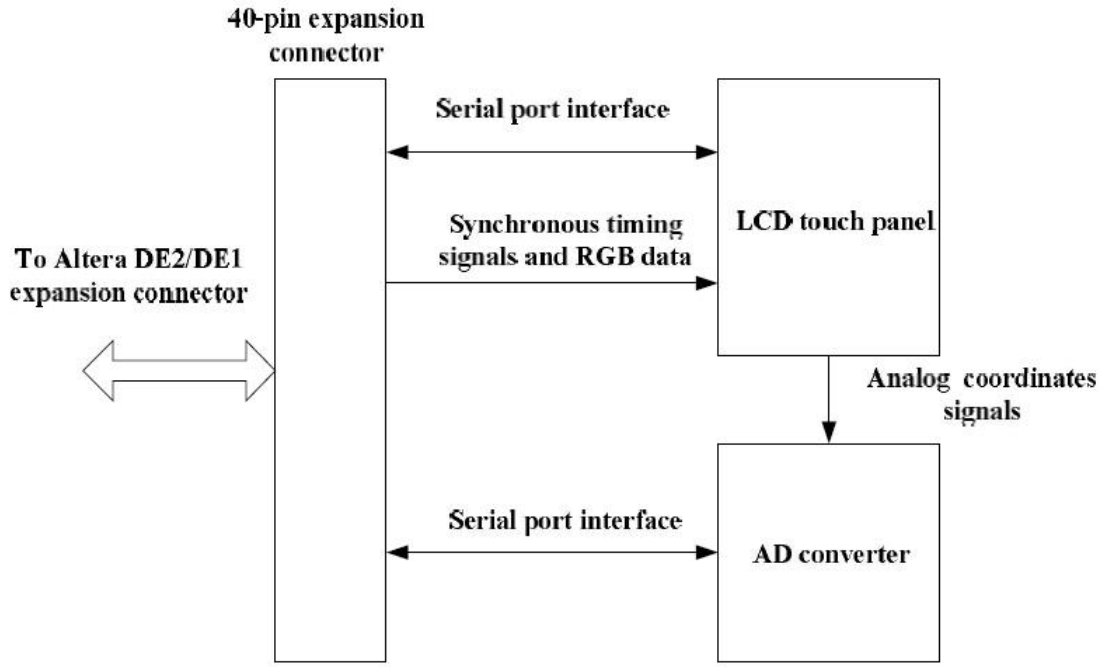


Figure 2.17 The block Diagram of the LTM[2]

CHAPTER 3

RESEARCH METHODOLOGY

This section discusses the methodology, the hardware and software used in the development of this project. The method of identify character used is Convolutional Neural Network (CNN) technique. It is a unique feed-forward multilayer perceptron trained in supervised mode by using a gradient descent back propagation learning algorithm. For instance, CNN is often implemented in image recognition system. CNN is studied and implemented a system which can recognize many digits and characters on touch terminal with classification accuracy that exceed 96% by using convolutional neural network technique. Next, CNN is developed on Field Programmable Gate Array (FPGA) by doing the hardware design using Verilog language. The Verilog code is compiled on the Altera Quartus 2 software. Then, connect touchpad or touchscreen to the FPGA board. GUI is set as a platform for users to enter the input such as character for verification and show the matched pattern or output on the screen.

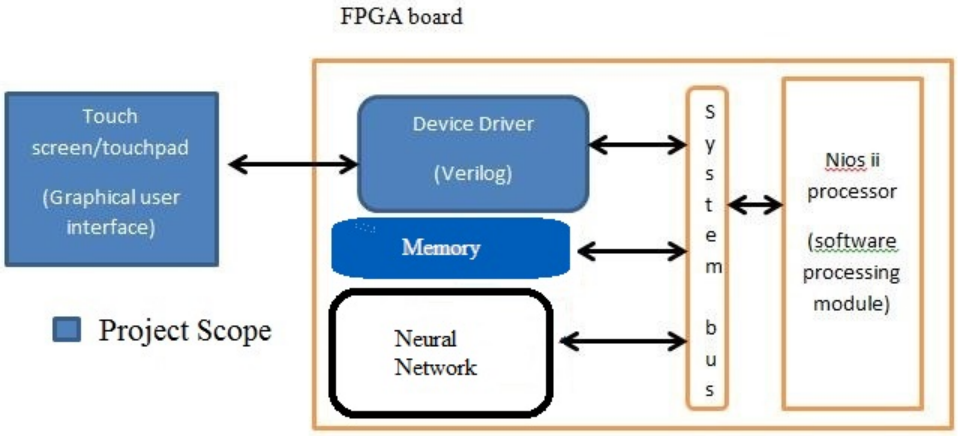


Figure 3.1 System Architecture

3.1 Process flow

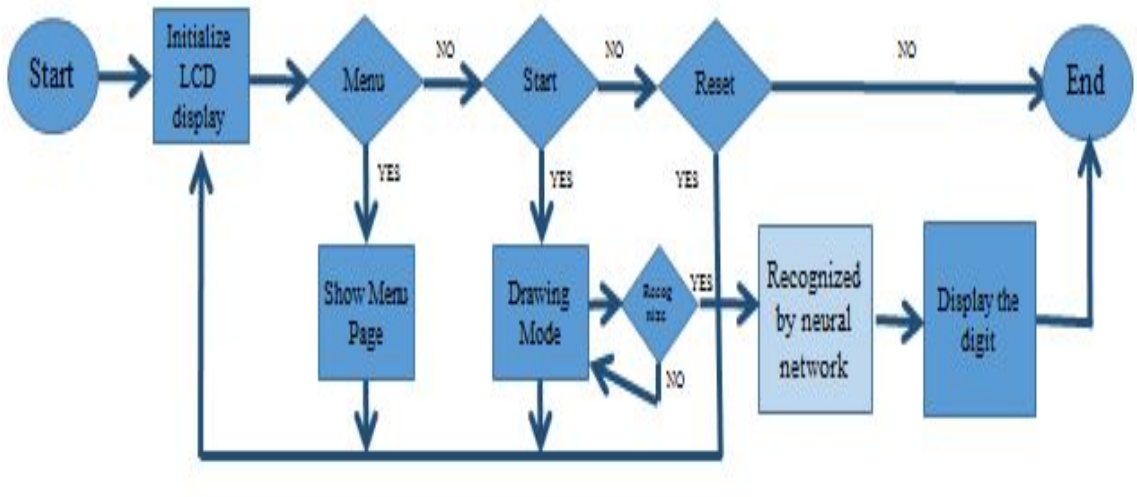


Figure 3.2 Process Flow

Figure 3.2 above depicts the process flow for this project. From the beginning of the project, LCD display is initialized. If the menu button is touched, the touch screen will display the menu page. Then, if the start button is touched, the system will proceed to drawing mode which allows the user to draw pattern on the touch screen. After that, if the recognize button is touched, the pattern which is drawn by user will be recognized by neural network. Then, the output digit is shown on the touch screen. In addition, the system will stay in drawing mode until the recognize button is touched. Besides, the system will initialize the LCD display if the reset push button is pressed.

3.2 Touch Screen Controller

The LCD and touch panel module on the LTM is equipped with an LCD driver integrated circuit (IC) to support three display resolutions and with functions of source driver, serial port interface, timing controller, and power supply circuits. To control these functions, users can use FPGA to configure the registers in the LCD driver IC via serial port interface. Also, there is an analog to digital converter (ADC) on the LTM to convert the analog X and Y coordinates of the touch point to digital data and output to FPGA through the serial port interface of the ADC. Both LCD driver IC and ADC serial port interfaces are connected to the FPGA via the 40-pin expansion header

and IDE cable.

Because of the limited number of I/O on the expansion header, the serial interfaces of the LCD driver IC and ADC need to share the same clock (ADC_DCLK) and chip enable (SCEN) signal I/O on the expansion header. To avoid both the serial port interfaces may interfere with each other when sharing the same clock and chip enable signals, the chip enable signal (CS), which is inputted into the ADC will come up with a logic inverter as shown in Figure 3.3. Users need to pay attention controlling the shared signals when designing the serial port interface controller. The specifications of the serial port interface of the LCD driver IC are described below.

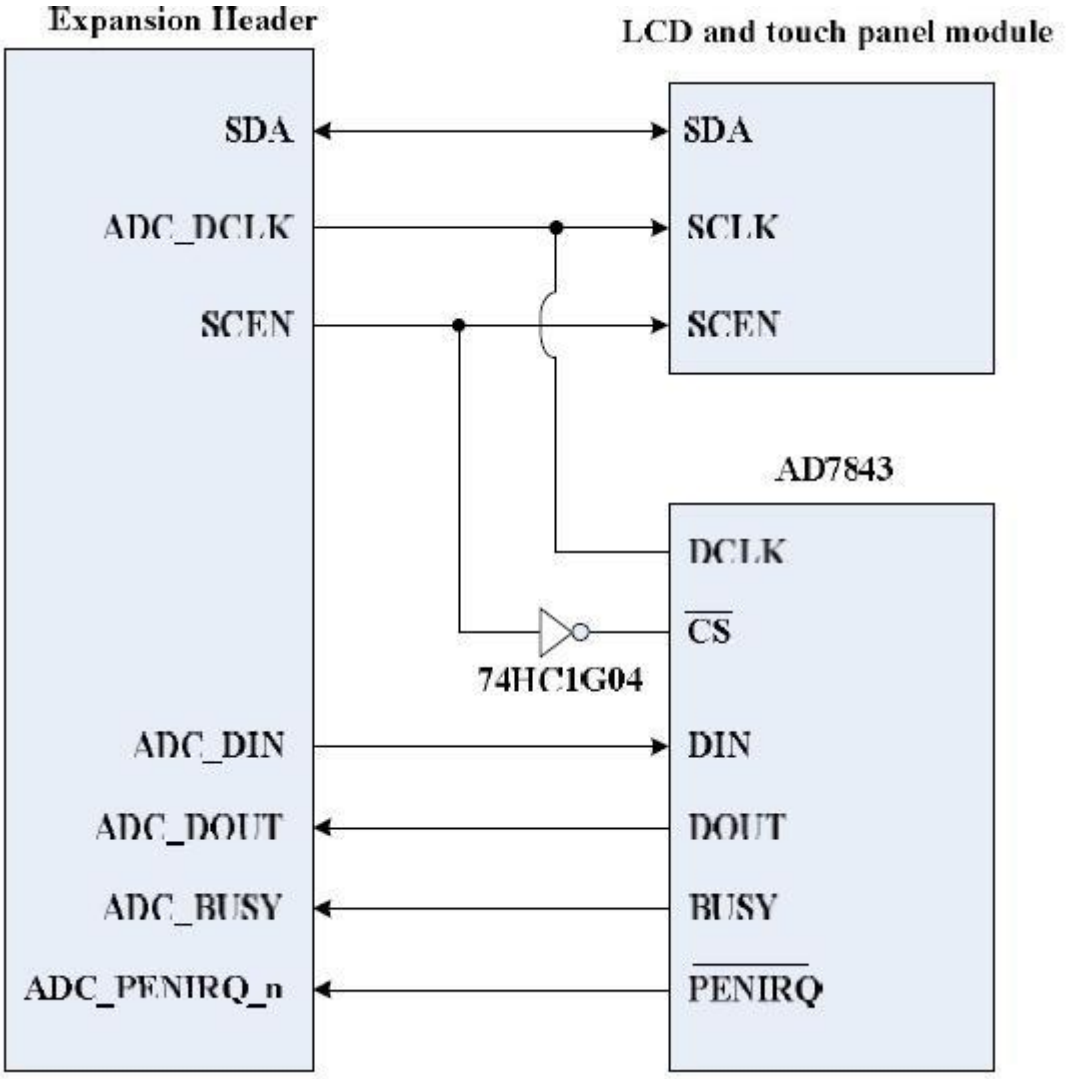


Figure 3.3 Serial interface of the LCD Touch Panel Module and AD7843

3.3 The Serial Interface of the Analog Digital (AD) Converter

This section will describe how to obtain the X/Y coordinates of the touch point from the AD converter. The touch screen is equipped with an Analog Devices AD7843 touchscreen digitizer chip. The AD7843is a 12-bit analog to digital converter (ADC) for digitizing x and y coordinates of touch points applied to the touch screen. To obtain the coordinate from the ADC, the first thing users need to do is monitor the interrupt signal ADC_PENIRQ_n outputted from the ADC. By connecting a pull high resistor, theADC_PENIRQ_n output remains high normally. When the touch screen connected to the ADC is touched via a pen or finger, the ADC_PENIRQ_n output goes low, initiating an interrupt to a FPGA that can then instruct a control word to be written to the ADC via the serial port interface.

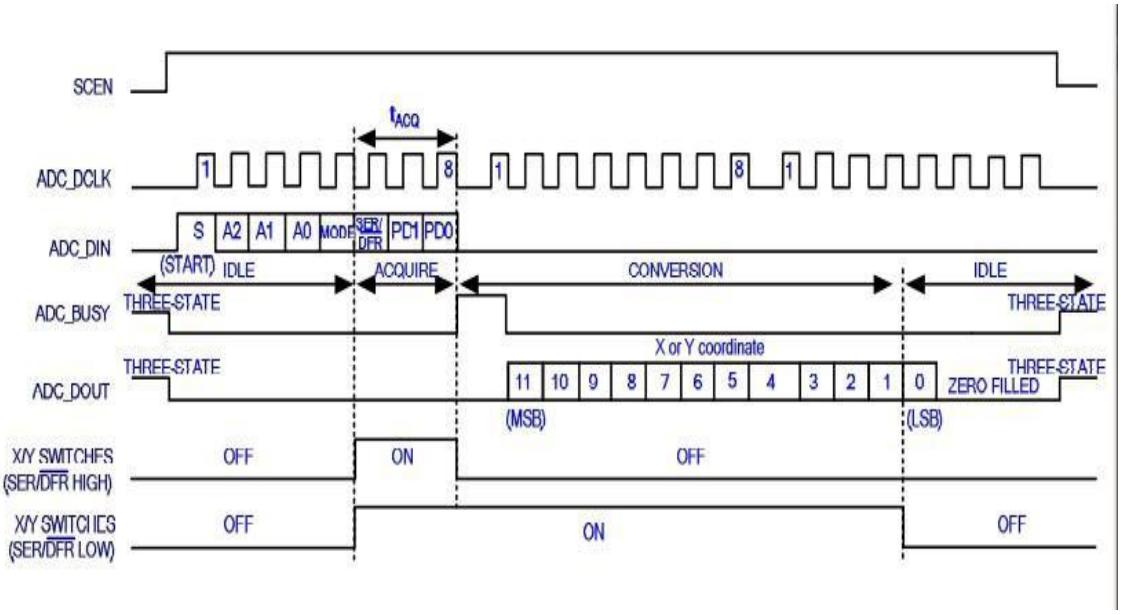


Figure 3.4 Conversion Timing of the Serial Port Interface

Figure 3.4 shows the typical operation of the serial interface of the ADC. The serial clock provides the conversion clock and also controls the transfer of information to and from the ADC. One complete conversion can be achieved with 24 ADC_DCLK cycles. The detailed behavior of the serial port interface can be found in the datasheet of the ADC. Note that the clock (ADC_DCLK) and chip enable signals (SCEN) of the serial port interface share the same signal I/O with LCD driver IC. Users should avoid controlling the LCD driver IC and ADC at the same time when designing the serial port interface controller. Also, because the chip enable signal (SCEN) input

to the ADC comes up with a logic inverter, the logic level of the SCEN should be inverse when it is used to control the ADC.ADC_DIN is pattern control signal of AD converter and ADC_DOUT is the coordinate of X or Y. Data can be transmitted when signal ADC_PENIRQ_n falls. ADC_BUSY controls the pattern of AD converter which enables to receive data when it keeps low.

3.4 Drawing Application

This section will discuss about the block diagram of the drawing application. Figure 3.5 shows the block diagram of the drawing application. Initially, the image can be written into flash memory in bitmap format using DE2 control panel. As soon as the bit stream is downloaded into the FPGA, the register values of the LCD driver IC used to control the LCD display function will be configured by the LCD_SPI_Controller block, which uses the serial port interface to communicate with the LCD driver IC. Meanwhile, the Flash_to_SDRAM_Controller block will read the RGB data of data stored in the Flash, and then write the data into SDRAM buffer. Accordingly, both the synchronous control signals and the data stored in the SDRAM buffer will be sent to the LTM via the LCD_Timing_Controller block. In addition, we can set up some drawings first on the touch screen by modify the code in LCD_Timing_Controller block. When the touch screen is touched, ADC_SPI_Controller block will detect the x and y coordinates of the touch point through the ADC serial port interface. Then the Touch_Point_Detector block detects whether the coordinates fit the range and send control signals to the LCD_Timing_Controller block to perform different functions like drawing and displaying numbers if the specific range is touched. In order to perform painting function, touched coordinates are sent to the LCD_Timing_Controller block continuously. Then, the different touched coordinates are stored in memory and split into four coordinates in order to create a small square dot with black color as shown on Figure 3.6. Algorithm 1 is the code for black dot creation, all the coordinates inside the range are black coloured because one pixel thickness is not clearly visible.

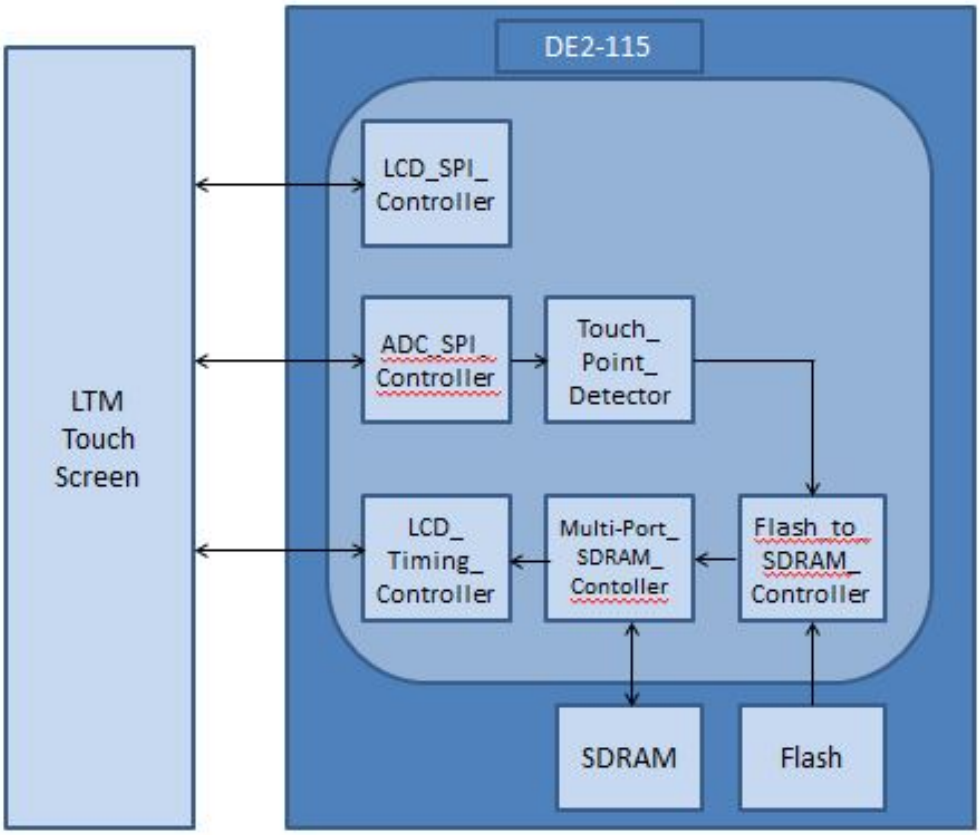


Figure 3.5 Block Diagram of the Drawing Application.

Algorithm 1 Code for Dot Creation.

```
else if(coord_x>Ycache[0]-number && coord_x<Ycache[0]+number &&
      coord_y>Xcache[0]-number &&coord_y<Xcache[0]+number && idraw_coord==1 && istop_coord==1)
         begin
         oHD   <= mhd;
         oVD   <= mvd;
         oDEN <= display_area;
            oLCD_R1 <= 10'h000;
            oLCD_G1 <= 10'h000;
            oLCD_B1 <= 10'h000;
            oLCD_R  <= oLCD_R1;
            oLCD_G  <= oLCD_G1;
            oLCD_B  <= oLCD_B1;
         end
```

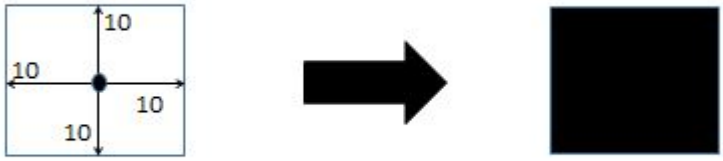


Figure 3.6 Square Dot with Black Colour Creation

3.5 Memory for Image Storage

A memory Initialization File (.mif) is an ASCII text file (with the extension .mif) that specifies the initial content of a memory block (CAM, RAM, or ROM), that is, the initial values for simulator. The Function of .Mif file in this project is loading data from memory and prototype on Altera FPGA board and then display on touch screen. Hexadecimal Intel-Format File (.hex) also can be used to provide memory initialization data. Algorithm 2 below shows the MATLAB code to perform bitmap image to raw data conversion. Figure 3.7 is a sample of .mif file created in Quartus II.

Algorithm 2 Bitmap to Raw Image Data Conversion.

MATLABcode:Img=imread('RESULT.BMP');BW=Img;R=BW(:,:,1);

Mif1.mif

Addr	+0	+1	+2	+3	+4	+5	+6	+7
0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0
56	0	0	0	0	0	0	0	0

Figure 3.7 Mif fle in Quartus II

3.6 Erosion Function

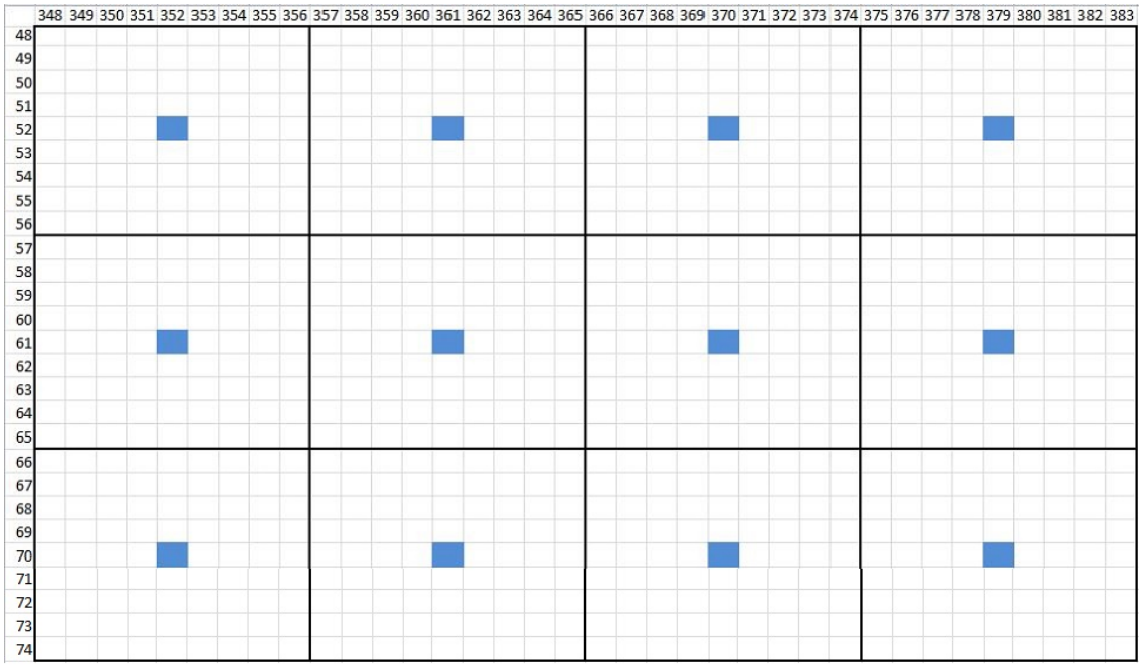


Figure 3.8 36x27 Pixels Image Data File.

Figure 3.8 shows the erosion. Erosion is a function that remove pixels which were near pixels that were off originally, thereby erode the pixels on the edges of the image. Above is the sample of erosion in 36x27 pixels on the touch screen, there are 288x288 pixels image as the drawing interface which allows the user to the draw digit. The convolutional neural network (CNN) block needs the input image to be 32x32 pixels in size. Therefore, the input image has to be eroded by a ratio of 9 times to produce the final 32x32 pixel image for the CNN block.

Hence, only one pixel of image is picked as the input image pixel in each of the 9x9 pixels of image. Figure 3.8 shows the method to choose one pixel of image within 9x9 pixels. The blue colour square which is located at the middle within 9x9 pixels of image on figure 3.8 is picked as the input image pixel. Figure 3.9 shows the sample of 32x32 input image for convolutional neural network block.

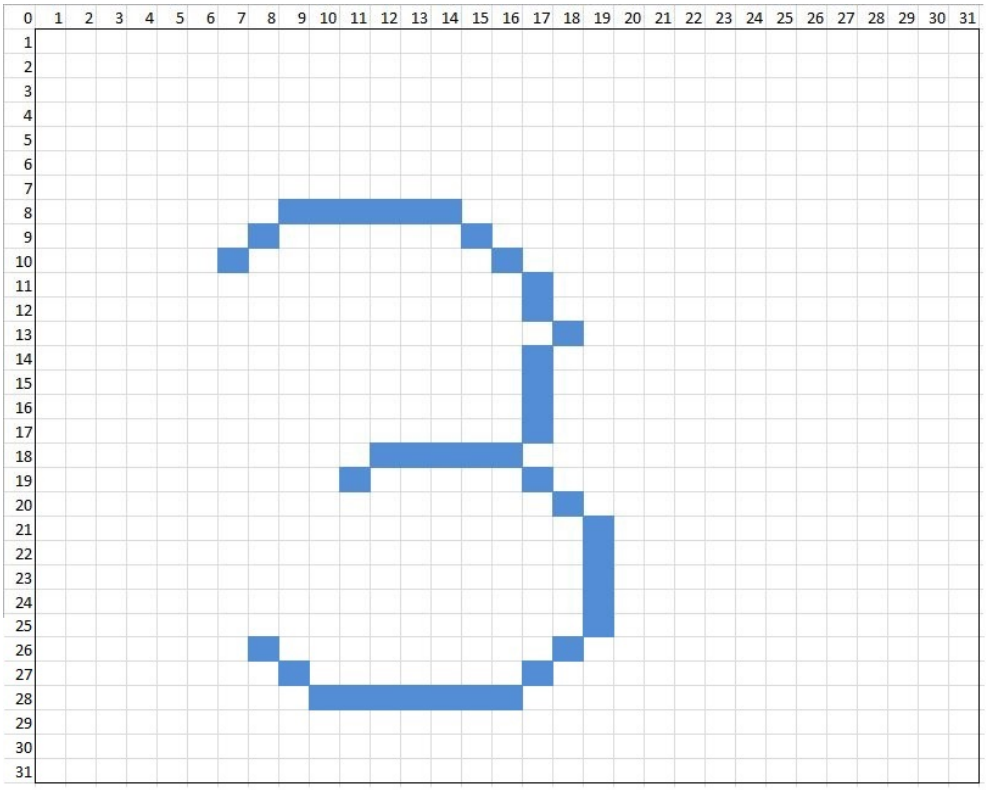


Figure 3.9 32x32 Pixels Image Data File.

CHAPTER 4

RESULT AND DISCUSSION

This section will discuss about the results of this project. The simulation waveforms and test results with DE2 board and touch panel are presented.

4.1 LCD module

The LCD module has built-in fonts and can be used to display text by sending appropriate commands to the display controller, which is called HD44780. Detailed information for using the display is available in its datasheet. A schematic diagram of the LCD module showing connections to the Cyclone II FPGA is given in Figure 4.1. The associated signals appear in Table 4.1. Figure 4.1 and Table 4.2 is the schematic diagram of the LCD module and character patterns with codes.

Table 4.1: Signal Assignments for the LCD Module.

Signal Name	Description
LCD_DATA[7:0]	LCD Data
LCD_RW	LCD Read/Write Select, 0 = Write, 1 = Read
LCD_EN	LCD Enable
LCD_RS	LCD Command/Data Select, 0 = Command, 1 = Data
LCD_ON	LCD Power ON/OFF
LCD_BLON	LCD Back Light ON/OFF

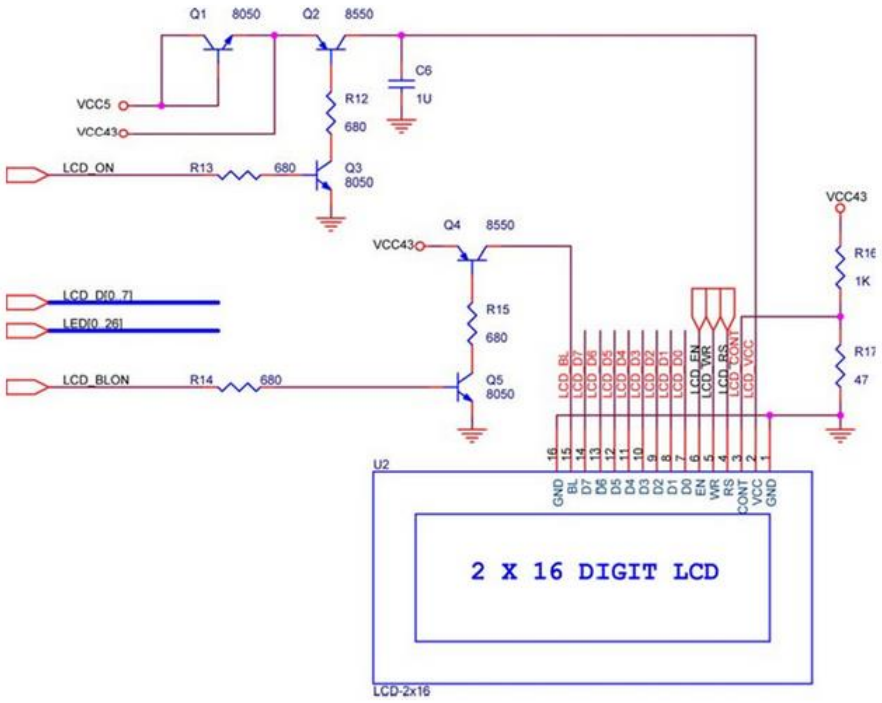


Figure 4.1 Schematic Diagram of the LCD Module

Table 4.2: Corresponded character codes and character patterns.

Table 4 Correspondence between Character Codes and Character Patterns (ROM Code: A00)

Lower 4 Bits \ Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)			0	@	P	`	p				ー	タ	ミ	α	p
xxxx0001	(2)		!	1	A	Q	a	q			。	ア	チ	ム	ä	q
xxxx0010	(3)		"	2	B	R	b	r			「	イ	ツ	メ	ß	θ
xxxx0011	(4)		#	3	C	S	c	s			」	ウ	テ	モ	ε	∞
xxxx0100	(5)		$	4	D	T	d	t			、	エ	ト	ヤ	μ	Ω
xxxx0101	(6)		%	5	E	U	e	u			・	オ	ナ	ユ	σ	Ü
xxxx0110	(7)		&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)		'	7	G	W	g	w			ァ	キ	ヌ	ラ	g	π
xxxx1000	(1)		(	8	H	X	h	x			ィ	ク	ネ	リ	√	x̄
xxxx1001	(2)		)	9	I	Y	i	y			ゥ	ケ	ノ	ル	⁻¹	y
xxxx1010	(3)		*	:	J	Z	j	z			ェ	コ	ハ	レ	j	千
xxxx1011	(4)		+	;	K	[	k	{			ォ	サ	ヒ	ロ	ˣ	万
xxxx1100	(5)		,	<	L	¥	l	|			ャ	シ	フ	ワ	¢	円
xxxx1101	(6)		−	=	M	]	m	}			ュ	ス	ヘ	ン	Ł	÷
xxxx1110	(7)		.	>	N	^	n	→			ョ	セ	ホ	゛	ñ	
xxxx1111	(8)		/	?	O	_	o	←			ッ	ソ	マ	°	ö	

Note: The user can specify any pattern for character-generator RAM.

After compilation in Quartus II, the generated .sof file is loaded into board. Different character patterns can be displayed on LCD. Name and date are displayed in the example shown in Figure 4.2.

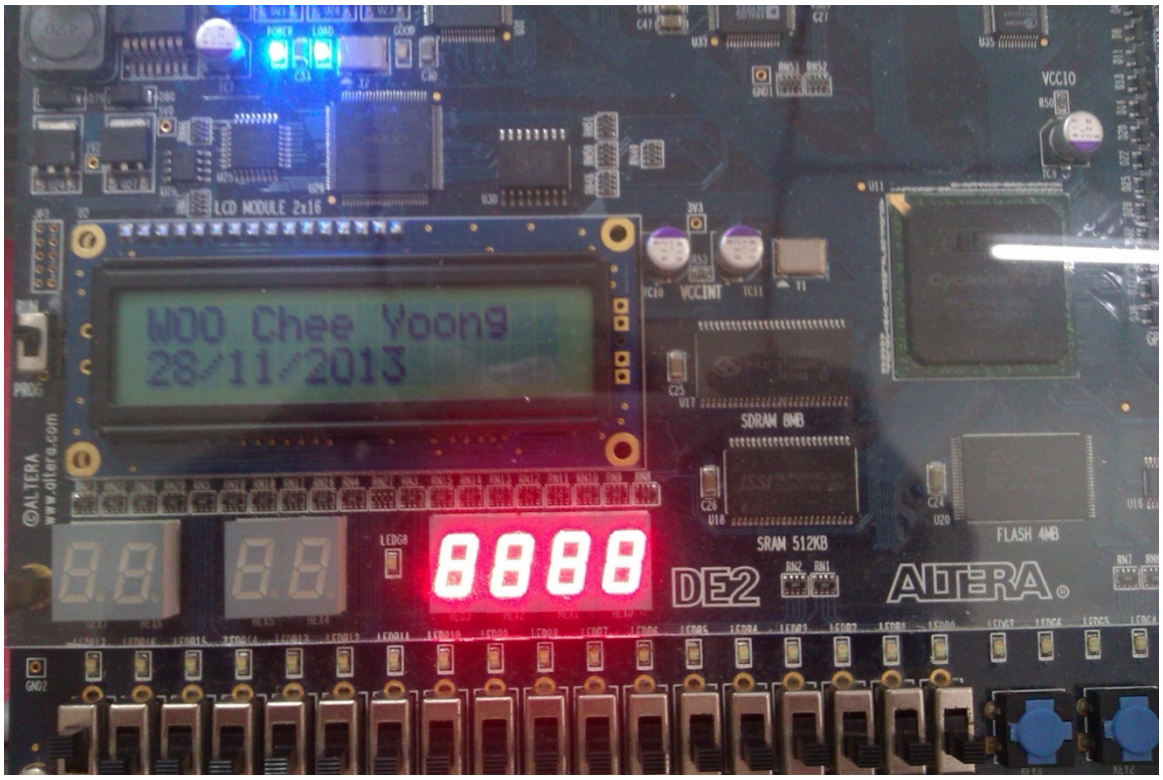


Figure 4.2 Result of LCD Module.

4.2 Memory Initialization File

The function of .Mif file is loading data from memory and prototype on altera FPGA board and then display on touch screen. In order to study the operation of .Mif file, a simple test is conducted and the assignment and result are shown in Table 4.3 and Figure 4.3.

Table 4.3: Memory Address Assignments

Addr	+0	+1	+2	+3	+4	+5	+6	+7
0	1	2	3	4	5	6	7	8
8	9	8	7	6	5	4	3	1

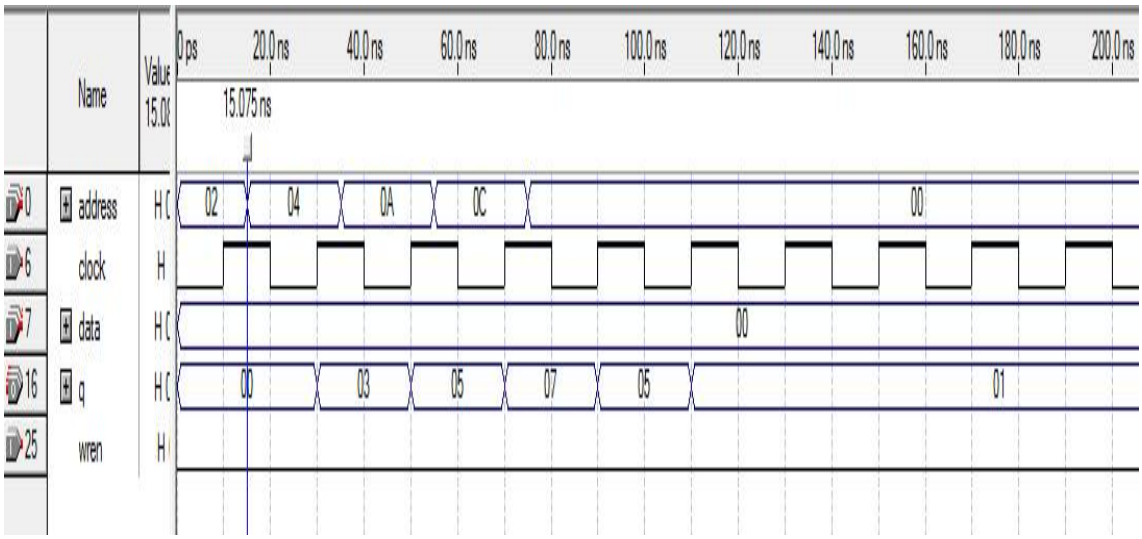


Figure 4.3 Vecor Waveform of Mif file.

Different values will be assigned into different memory addresses randomly, when certain addresses is assigned as inputs, the corresponding value stored in memory will be shown at the waveform q which is output.

4.3 Touch Screen module

This section is to test the function of the TRDB-LTM module.

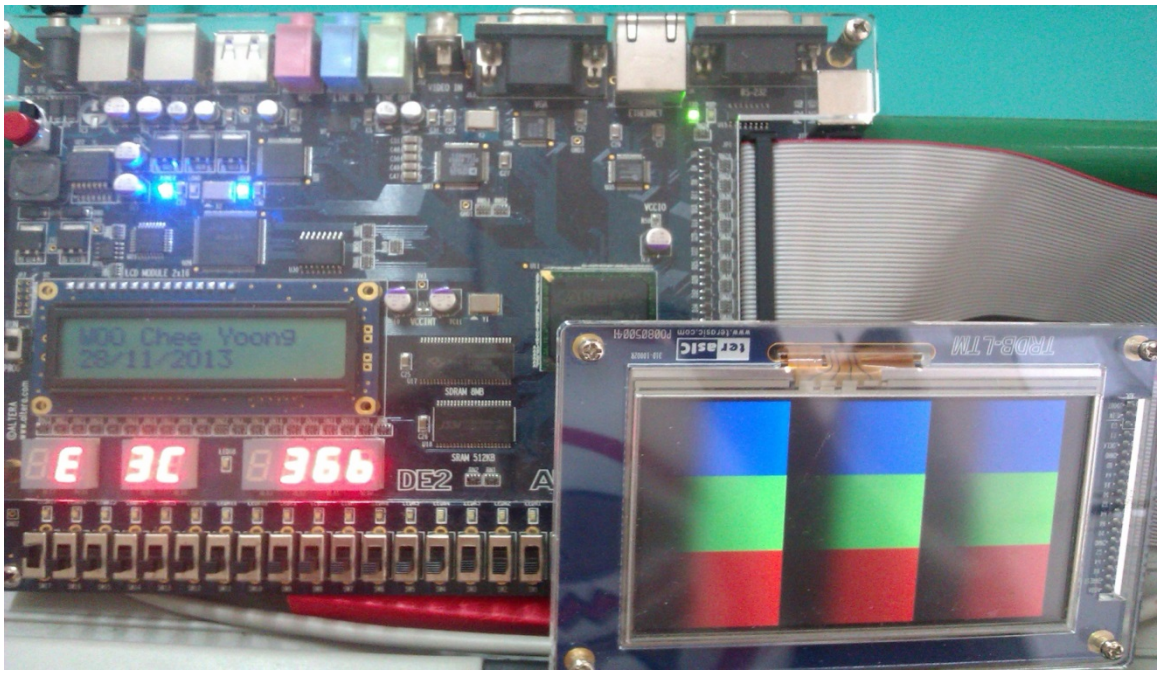


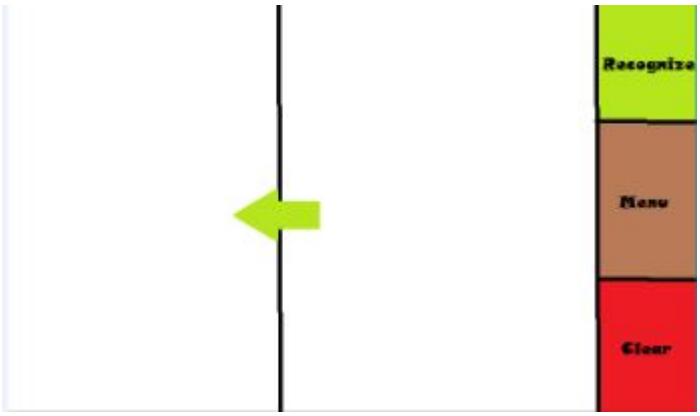
Figure 4.4 TRDB-LTM Touch Screen and DE2-35 Board.

Figure 4.4 shows that the touch panel connected to DE2 board. When certain coordinate is touched by stylus pen or finger, the touch point detected will be converted through the AD converter which is build inside the touch panel. This AD converter will convert the touch point detected to digital data and show the X and Y coordinates on the seven segment LEDs. The three digits on the left are showing the vertical coordinate of the touch point detected whereas another three digits are showing the horizontal coordinate. In conclusion, the touch panel is functioning.

4.4 Drawing Application

Figure 4.8 shows the bitmap picture that is loaded into the flash memory of DE2-115 board via DE2-115 control panel.

(a) Initialization Page



(b) Drawing Mode

(c) Menu

Figure 4.5 480x1440 Bitmap Image with (a)Initialization page, (b)Drawing Mode and (c)Menu

Initially, the LCD display will be initialized by showing the UTM page. Then, if the menu button is pressed, the LCD display will change the menu page and the start button is for enter the drawing mode. In addition, the user can press recognized to recognize digit and press the reset button on DE2-115 board to clear the drawing and

the memory stored in RAM. Figure 4.6 shows the compilation report.

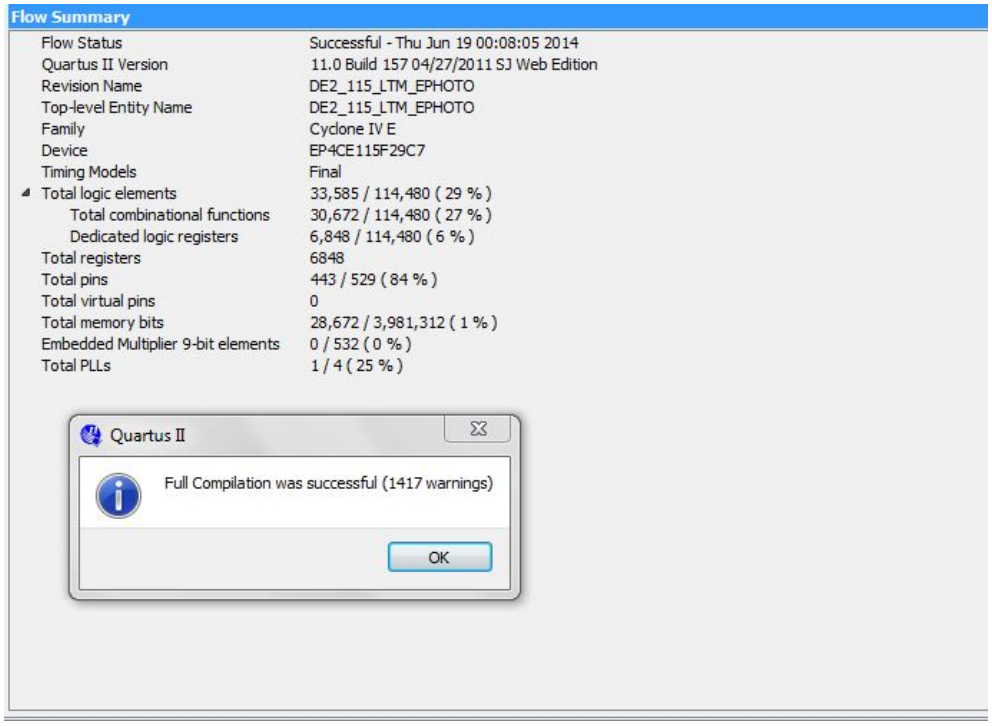


Figure 4.6 Compilation Report

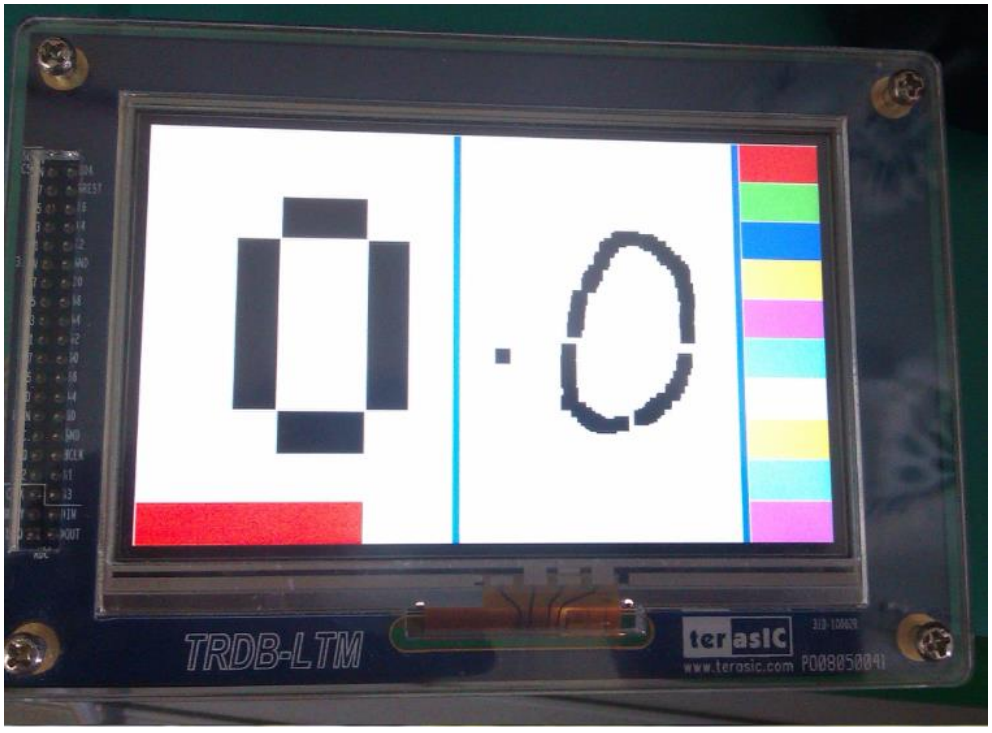
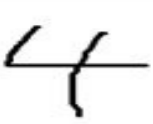
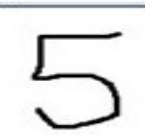
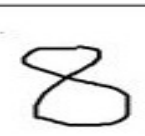
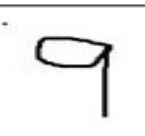


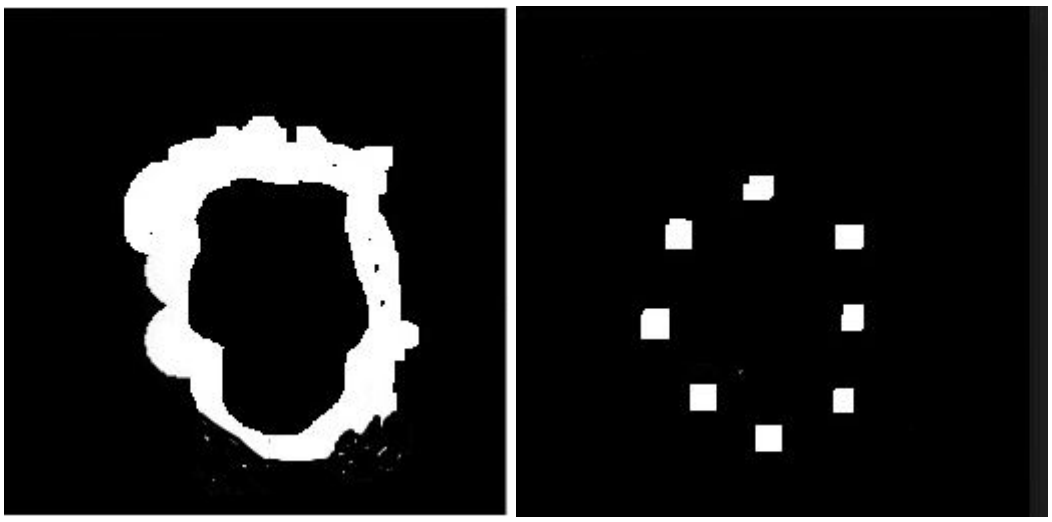
Figure 4.7 Digit Recognition System on Touch Screen.

After the program is downloaded into DE2 board, the touch screen is activated as shown on Figure 4.7. The digit shown on left hand side is drawn from the LCD_Timing_Controller block whereas the digit on right hand side is drawn using a stylus pen. The colors shown on the touch screen indicates there are a lot of colors can be drawn by sending different value of RGB. On the DE2, KEY0 is set as reset and the seven segment LEDs (HEX0 to HEX6) are displaying the x and y coordinates (from 000 to FFF) of the touch point detected. Initially, DE2-35 is using in this application. However, there are unstablity problems that occurred after the program is downloaded. The painting system not as smooth as predicted, this because of the design problem or it's has reached the limit of the touch screen. Then, the hang problem is solved by using DE2-115 instead. Table 4.4 shows the inputs digit and the outputs that can be recognized by convolutional network.

Table 4.4: Inputs Digit and Outputs

Input images	output
	
	
	
	

4.5 Image Erosion



(a) Before Erode **(b)** After Erode

Figure 4.8 Image Erosion.

The erode function is morphological operations, erosion removes pixels on object boundaries. Below is an example to show erode image function. Some of the pixels on Figure 4.8 (a) is removed to simplified the image for recognition purpose. In this project, erosion is applied on binary image that only has two possible values for each pixel, white colour is sets as zero whereas black colour is set as one.

CHAPTER 5

PROJECT MANAGEMENT

The objective of project management is to achieve all project goals with effective project planning, organizing, and controlling resource within a specified time period. The primary constraints in this project are the research scope, research time, research budget and human resource to perform the required activity. Based on the stated constraints, project schedule had been tabulated on Gantt chart which it gives a clear guideline in time management of this project.

5.1 Project Schedule

Table 5.1 shows project Gantt chart for semester one. In fact, 4 th year problem based solving lab is started earlier than the date of announcing supervisor in charge for project. Hence, familiarization on Altera DE1/2 board has done early along with 4 th year lab. The study of the related works is start from semester start to semester end.

Table 5.1: Gantt Chart Semester 1

No	Activities/Tasks	Weeks															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Literature Review																
2	Study on Verilog language																
3	Study on DE2 boadrd																
4	Study on TRDB-LTM touch panel																
5	Study on Quatus II																
6	Loading program on board																
7	Loding program on touch panel																
8	Presentation																

Table 5.2: Gantt Chart Semester 2

No	Activities/Tasks	Weeks															
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	Literature Review																
2	Interfacing TRDB-LTM touch panel																
3	Decision making module																
4	Project Testing and Verification																
5	Implementing Neural Network on FPGA																
6	Project testing and Verification																

Table 5.2 shows project Gantt chart for semester two. In contrast to Gantt chart semester one, Gantt chart in second semester showed an expected long delay in some task at end of the project schedule. The delay on implementing neural network on FPGA is due to the lack of DE2-115 Development Board in University and the incorrect recognized results by CNN.

CHAPTER 6

CONCLUSION

This chapter discusses the conclusion and future works from this project. The objective of this project is to develop a drawing application on TRDB-LTM touch screen by purely hardware implementation and perform image erosion 288x288 pixels to 32x32 pixels conversion as an input for convolutional neural network block.

In this project, there is no proper device driver to interface between the touch panel and the FPGA board to perform drawing application.Then, another problem is that there is no graphical user interface (GUI) for the touch panel. Furthermore, most of the character recognition systems are software based.

Hence, the objectives of the project is to build a suitable device driver which enables the TRDB-LTM touch panel to communicate properly with the Altera DE2 board. After that, design a GUI on the touch screen which allows the user to draw digits, gives different instructions on desired touch point coordinates and displays proper image file result and text. Then, build a pure hardware implementation for the system with high speed response, high security and stand alone device.

This project used DE2-115 Development and Education Board and TRDB_LTM (LTM) Touch Panel to carry out the drawing application and processing the inputs. Altera Quartus II is the software platform and the code used is verilog code. In addition, neural network is not included in the project scope.

There are some limitations on this project. The drawing process is not as smooth as expect, the number of black dots for drawing is limited due to the limitation of logic element for DE2-115 board and noise could be occurred randomly during the drawing process.

Overall, the project had been successfully developed which fulfilled all the objective and requirements. The system is working in real time and performs drawing application and an user-friendly GUI is created by hardware implementation.

Furthermore, this project can be integrated with convolutional neural network to create an pattern recognition system. This system is high speed, strong in security and precise due to the strengths of hardware design and convolutional neural network. In addition, the applications of this pattern recognition system are e-cheque banking, children educational board and any security systems.

REFERENCES

1. Utm thesis manual, 2007.
2. Trdb-ltm touch screen user manual. *Terasic Technologies*, 2009.
3. Veek-mt user manual. *Terasic Technologies*, 2011.
4. Keith Adams and Ole Agesen. A comparison of software and hardware techniques for x86 virtualization. *ACM Sigplan Notices*, 41(11):2–13, 2006.
5. Salvatore Amaro, Alessandro Viggiano, Anna Di Costanzo, Ida Madeo, Andrea Viggiano, Maria Ena Baccari, Elena Marchitelli, Maddalena Raia, Emanuela Viggiano, Sunil Deepak, et al. Kalèdo, a new educational board-game, gives nutritional rudiments and encourages healthy eating in children: a pilot cluster randomized trial. *European journal of pediatrics*, 165(9):630–635, 2006.
6. Derrick A Bell Jr. Brown v. board of education and the interest-convergence dilemma. *Harv. L. Rev.*, 93:518, 1979.
7. James K Chan. Educational/board game apparatus, February 18 1992. US Patent 5,088,928.
8. Scanner Chen. Method for controlling united home security system, May 9 2000. US Patent 6,060,994.
9. Tien-Fu Chen and Jean-Loup Baer. A performance study of software and hardware data prefetching schemes. In *Computer Architecture, 1994., Proceedings the 21st Annual International Symposium on*, pages 223–232. IEEE, 1994.
10. Bhavya Daya. Rapid prototyping of embedded systems using field programmable gate arrays. *Bachelor of Science In Electrical Engineering, Spring*, 2009.
11. Stephen A Edwards. Experiences teaching an fpga-based embedded systems class. *ACM SIGBED Review*, 2(4):56–62, 2005.
12. Jeong Kim and John Zimmerman. Cherish: smart digital photo frames for sharing social narratives at home. In *CHI'06 Extended Abstracts on Human*

Factors in Computing Systems, pages 953–958. ACM, 2006.

13. Hamid Mahmoodi, Arturo Montoya, Joie du Franco, Chris Rodriguez, Jose Carrillo, Ankita Goel, Cheng Chen, Amelito G Enriquez, Hao Jiang, Wenshen Pong, et al. Hands-on teaching of embedded systems design using fpga-based tpad development kit. In *Interdisciplinary Engineering Design Education Conference (IEDEC), 2012 2nd*, pages 1–6. IEEE, 2012.

14. Thomas Eriksson Martin Olsson, Samuel SkÃ¥nberg. Digital photo frame. *EDA385 - Design of Embedded Systems Advanced Course, LP1 Ht09*, 2009.

15. Vijayakrishnan Pasupathinathan, Josef Pieprzyk, and Huaxiong Wang. Privacy enhanced electronic cheque system. In *CEC*, pages 431–434, 2005.

16. David A Patterson and John L Hennessy. *Computer organization and design: the hardware/software interface*. Newnes, 2013.

17. Leo Paulose and Phani Teja Cheedella. *Enhancement of Digital Photo Frame Capabilities With Dedicated Hardware*. PhD thesis, 2012.

18. Ronald E Pyle. Home security system, May 1 1984. US Patent 4,446,454.

19. Thomas Schumann. Advanced vlsi design methodologies for emerging industrial multimedia and communication applications.

20. OR Vincent and O Folorunso. A descriptive algorithm for sobel image edge detection. In *Proceedings of Informing Science & IT Education Conference (InSITE)*, pages 97–107, 2009.

21. Si-Ming Xia, Xiao-Le Xu, Liang Qin, and Chang-Hua Liu. Galaxian game on altera de2-115 fpga architecture.

22. Ran Zheng Younggyun Cho Xiang Zhou, Hao Zheng. Touch pong. *CSEE 48409, Columbia University*, 2009.

23. C.M.Wu C.Y Chen.Y.S.Lin Y.Y Chang, C.M.Huang. *A Modularized FPGA-Based Embedded System Development Platform, National Central University,*. National Central University, 2010.

APPENDIX A

DRAWING APPLICATION

```
module lcd_timing_controller
( iCLK,          // LCD display clock
iRST_n,          // systen reset
// SDRAM SIDE
iREAD_DATA, // R and G  color data form sdram
//iREAD_DATA2,// B color data form sdram
oREAD_SDRAM_EN,// read sdram data control signal
//LCD SIDE
oHD,     // LCD Horizontal sync
oVD,     // LCD Vertical sync
oDEN,    // LCD Data Enable
oLCD_R, // LCD Red color data
oLCD_G,    // LCD Green color data
oLCD_B, // LCD Blue color data
idraw_coord,
istop_coord,
iz_coord,
io_coord,
itwo_coord,
ithree_coord,
ifour_coord,
ifive_coord,
isix_coord,
iseven_coord,
ieight_coord,
inine_coord,
imenu_coord,
iX_COORD,
iY_COORD,
```

```
i_coord,
);
parameter H_LINE = 12'h420;
parameter V_LINE = 12'h20d;
parameter Hsync_Blank = 8'hd8;
parameter Hsync_Front_Porch = 8'h28;
parameter Vertical_Back_Porch = 8'h23;
parameter Vertical_Front_Porch = 4'ha;
input                   iCLK;
input                   iRST_n;
input   [31:0]  iREAD_DATA; //
input   [15:0]  iREAD_DATA2;
input           idraw_coord;
input                   istop_coord;
input           iz_coord;
input                   io_coord;
input           itwo_coord;
input                   ithree_coord;
input           ifour_coord;
input                   ifive_coord;
input           isix_coord;
input                   iseven_coord;
input           ieight_coord;
input                   inine_coord;
input                   imenu_coord;
input                   i_coord;
input [11:0]    iX_COORD;
input [11:0]    iY_COORD;
output  oREAD_SDRAM_EN;
output  [7:0]   oLCD_R;
output  [7:0]   oLCD_G;
output  [7:0]   oLCD_B;
output                  oHD;
output                  oVD;
output                  oDEN;
reg             [10:0]  x_cnt;
reg             [9:0]   y_cnt;
reg             [9:0]  coord_x;
reg             [9:0]  coord_y;
```

```
wire    [7:0]   read_red;
wire    [7:0]   read_green;
wire    [7:0]   read_blue;
wire                    display_area;
wire                    oREAD_SDRAM_EN;
reg                             mhd;
reg                             mvd;
reg                             oHD;
reg                             oVD;
reg                             oDEN;
reg             [7:0]   oLCD_R;
reg             [7:0]   oLCD_G;
reg             [7:0]   oLCD_B;
wire            [11:0]  mx_coord;
wire            [11:0]  my_coord;
reg             [11:0]Xcache[0:memsize-1];
reg             [11:0]Ycache[0:memsize-1];
reg             [348:636]image[48:336]; //stored image file
reg             [0:31]in_image[0:31];
parameter memsize=512;
parameter number=10;
reg             [9:0] k,l,m,n,c,d,e,f;
integer index;
wire [0:32] temp [0:32];

// This signal control reading data form SDRAM ,
if high read color data form sdram  .
assign  oREAD_SDRAM_EN = (
(x_cnt>Hsync_Blank-2)&&
(x_cnt<(H_LINE-Hsync_Front_Porch-1))&&
(y_cnt>(Vertical_Back_Porch-1))&&
(y_cnt<(V_LINE - Vertical_Front_Porch))
)?  1'b1 : 1'b0;

// This signal indicate the lcd display area .
assign  display_area = ((x_cnt>(Hsync_Blank-1)
&& //>215 (x_cnt<(H_LINE-Hsync_Front_Porch))&&
//< 1016 (y_cnt>(Vertical_Back_Porch-1))&&
(y_cnt<(V_LINE - Vertical_Front_Porch))
```

```
))  ? 1'b1 : 1'b0;
assign  read_red        =
display_area ? iREAD_DATA[31:24] : 8'b0;
assign  read_green      =
display_area ? iREAD_DATA[23:16]: 8'b0;
assign  read_blue       =
display_area ? iREAD_DATA[15:8] : 8'b0;
assign  mx_coord = (iX_COORD*100/853);
assign  my_coord = (iY_COORD*100/512);

always@(posedge iCLK or negedge iRST_n)
        begin
if (!iRST_n)
        begin
        x_cnt <= 11'd0;
                        mhd  <= 1'd0;
                        end
        else if (x_cnt == (H_LINE-1))
        begin
        x_cnt <= 11'd0;
                mhd  <= 1'd0;
        end
else
begin
        x_cnt <= x_cnt + 11'd1;
                mhd  <= 1'd1;
        end
        end
always@(posedge iCLK or negedge iRST_n)
        begin
if (!iRST_n)
                y_cnt <= 10'd0;
        else if (x_cnt == (H_LINE-1))
        begin
        if (y_cnt == (V_LINE-1))
                        y_cnt <= 10'd0;
                        else
        y_cnt <= y_cnt + 10'd1;
                end
```

```
end

always@(posedge iCLK  or negedge iRST_n)
begin            if (!iRST_n)
                 mvd  <= 1'b1;
else if (y_cnt == 10'd0)
                 mvd  <= 1'b0;
        else
mvd  <= 1'b1;
        end
reg [7:0] oLCD_R1;
reg [7:0] oLCD_G1;
reg [7:0] oLCD_B1;
always@(posedge iCLK  or negedge iRST_n)
begin
        if(!iRST_n)
                 begin
k<=0;
for(l=0; l<384 ; l=l+1)
begin
Xcache[l]<=0;
Ycache[l]<=0;
end
for(c=48;c<336;c=c+1)
begin
for(e=348;e<636;e=e+1)
begin
image[c][e]<=0;
end
end
end

else if (i_coord && (my_coord>300
&&my_coord<684))
begin
k<=k+1;
Xcache[k]=mx_coord;
Ycache[k]=my_coord;
image[mx_coord][my_coord]<=1;
```

```
end
end
always@(posedge iCLK or negedge iRST_n)
        begin
        if (!iRST_n)
                begin
                        oHD     <= 1'd0;
                        oVD     <= 1'd0;
                        oDEN <= 1'd0;
                        oLCD_R <= 8'd0;
                        oLCD_G <= 8'd0;
                        oLCD_B <= 8'd0;
        end
else if(coord_y>150 && coord_y<350 && coord_x>50
&& coord_x<100 && iz_coord ==1 && istop_coord==1)
//0
                begin
                        oHD     <= mhd;
                        oVD     <= mvd;
                        oDEN <= display_area;
                   oLCD_R1 <= 10'h000;
                        oLCD_G1 <= 10'h000;
                        oLCD_B1 <= 10'h000;
                        oLCD_R  <= oLCD_R1;
                        oLCD_G  <= oLCD_G1;
                        oLCD_B  <= oLCD_B1;
                end
else if(coord_y>350 && coord_y<400 &&
coord_x>100 && coord_x<200
&& iz_coord ==1&& istop_coord==1) //0
                        begin
                        oHD     <= mhd;
                        oVD     <= mvd;
                        oDEN <= display_area;
                        oLCD_R1 <= 10'h000;
                        oLCD_G1 <= 10'h000;
                        oLCD_B1 <= 10'h000;
                        oLCD_R  <= oLCD_R1;
                        oLCD_G  <= oLCD_G1;
```

```
                        oLCD_B   <= oLCD_B1;
                    end
 else if(coord_y>150 && coord_y<350
&& coord_x>200 && coord_x<250
&& iz_coord ==1&& istop_coord==1) //0
                    begin
                    oHD      <= mhd;
                    oVD      <= mvd;
                    oDEN <= display_area;
                    oLCD_R1 <= 10'h000;
                    oLCD_G1 <= 10'h000;
                    oLCD_B1 <= 10'h000;
                    oLCD_R   <= oLCD_R1;
                    oLCD_G   <= oLCD_G1;
                    oLCD_B   <= oLCD_B1;
                    end
 else if(coord_y>100 && coord_y<150
&& coord_x>100 && coord_x<200
&& iz_coord ==1&& istop_coord==1) //0
          begin
                    oHD      <= mhd;
                    oVD      <= mvd;
                    oDEN <= display_area;
                    oLCD_R1 <= 10'h000;
                    oLCD_G1 <= 10'h000;
                    oLCD_B1 <= 10'h000;
                    oLCD_R   <= oLCD_R1;
                    oLCD_G   <= oLCD_G1;
                    oLCD_B   <= oLCD_B1;
 end
 else if(coord_y>100 && coord_y<400
&& coord_x>175 && coord_x<225
&& io_coord ==1&& istop_coord==1) //1
                    begin
                    oHD      <= mhd;
                    oVD      <= mvd;
                    oDEN <= display_area;
                    oLCD_R1 <= 10'h000;
                    oLCD_G1 <= 10'h000;
```

```
                    oLCD_B1 <= 10'h000;
                    oLCD_R  <= oLCD_R1;
                    oLCD_G  <= oLCD_G1;
                    oLCD_B  <= oLCD_B1;
                    end
else if(coord_y>350 && coord_y<400 &&
coord_x>25 && coord_x<225 &&
(itwo_coord==1||ithree_coord==1
||ifive_coord ==1||isix_coord==1||
iseven_coord==1||ieight_coord==1||
inine_coord==1 )&& istop_coord==1)
//2,3,5,6,7,8,9
                    begin
                    oHD     <= mhd;
                    oVD     <= mvd;
                    oDEN <= display_area;
                    oLCD_R1 <= 10'h000;
                    oLCD_G1 <= 10'h000;
                    oLCD_B1 <= 10'h000;
                    oLCD_R  <= oLCD_R1;
                    oLCD_G  <= oLCD_G1;
                    oLCD_B  <= oLCD_B1;
                    end
else if(coord_y>220 && coord_y<400 &&
coord_x>175 && coord_x<225 &&
(itwo_coord ==1|| ithree_coord ==1
||ifour_coord ==1 ||iseven_coord==1||
ieight_coord==1||inine_coord==1)
&& istop_coord==1)//2,3,7,8,9
          begin
          oHD     <= mhd;
          oVD     <= mvd;
          oDEN <= display_area;
          oLCD_R1 <= 10'h000;
          oLCD_G1 <= 10'h000;
          oLCD_B1 <= 10'h000;
          oLCD_R  <= oLCD_R1;
          oLCD_G  <= oLCD_G1;
          oLCD_B  <= oLCD_B1;
```

```
        end
else if(coord_y>200 && coord_y<250
&& coord_x>25 && coord_x<225 &&
(itwo_coord ==1|| ithree_coord ==1
||ifour_coord ==1||ifive_coord==1
||isix_coord==1||ieight_coord==1
||inine_coord==1)&&istop_coord==1)
//2,5,6,8,9
begin
        oHD     <= mhd;
        oVD     <= mvd;
        oDEN <= display_area;
        oLCD_R1 <= 10'h000;
        oLCD_G1 <= 10'h000;
        oLCD_B1 <= 10'h000;
        oLCD_R  <= oLCD_R1;
        oLCD_G  <= oLCD_G1;
        oLCD_B  <= oLCD_B1;
        end
else if(coord_y>100 && coord_y<200
&& coord_x>25 && coord_x<75 &&
(itwo_coord ==1||ieight_coord==1)
        && istop_coord==1) //2,8
begin
        oHD     <= mhd;
        oVD     <= mvd;
        oDEN <= display_area;
        oLCD_R1 <= 10'h000;
        oLCD_G1 <= 10'h000;
        oLCD_B1 <= 10'h000;
        oLCD_R  <= oLCD_R1;
        oLCD_G  <= oLCD_G1;
        oLCD_B  <= oLCD_B1;
end
else if(coord_y>50 && coord_y<100 &&
coord_x>25 && coord_x<225 &&
(itwo_coord ==1|| ithree_coord ==1
||ifive_coord==1||isix_coord==1||
ieight_coord==1||inine_coord==1)
```

```
&& istop_coord==1) //2,5,6,8,9
        begin
        oHD     <= mhd;
        oVD     <= mvd;
        oDEN <= display_area;
        oLCD_R1 <= 10'h000;
        oLCD_G1 <= 10'h000;
        oLCD_B1 <= 10'h000;
        oLCD_R  <= oLCD_R1;
        oLCD_G  <= oLCD_G1;
        oLCD_B  <= oLCD_B1;
        end
 else if(coord_y>50 && coord_y<250
&& coord_x>175 && coord_x<225 &&
(ithree_coord==1||ifour_coord==1||
ifive_coord==1||isix_coord==1||
ieight_coord==1||iseven_coord==1
||inine_coord==1)&&istop_coord==1)//3,5,6,7,8,9
        begin
                oHD     <= mhd;
                oVD     <= mvd;
                oDEN <= display_area;
                oLCD_R1 <= 10'h000;
                oLCD_G1 <= 10'h000;
                oLCD_B1 <= 10'h000;
                oLCD_R  <= oLCD_R1;
                oLCD_G  <= oLCD_G1;
                oLCD_B  <= oLCD_B1;
                end
        else if(coord_y>200 && coord_y<400
&& coord_x>25 && coord_x<75 &&
(ifour_coord ==1 ||ifive_coord ==1
||isix_coord ==1||ieight_coord==1||
inine_coord==1)&&
istop_coord==1) //4,5,6,8,9
        begin
        oHD     <= mhd;
        oVD     <= mvd;
        oDEN <= display_area;
```

```
        oLCD_R1 <= 10'h000;
        oLCD_G1 <= 10'h000;
        oLCD_B1 <= 10'h000;
        oLCD_R  <= oLCD_R1;
        oLCD_G  <= oLCD_G1;
        oLCD_B  <= oLCD_B1;
        end
else if(coord_y>50 && coord_y<250
&& coord_x>25 && coord_x<75 &&
(isix_coord ==1||ieight_coord==1)
        && istop_coord==1) //6,8
        begin
        oHD     <= mhd;
        oVD     <= mvd;
        oDEN <= display_area;
        oLCD_R1 <= 10'h000;
        oLCD_G1 <= 10'h000;
        oLCD_B1 <= 10'h000;
        oLCD_R  <= oLCD_R1;
        oLCD_G  <= oLCD_G1;
        oLCD_B  <= oLCD_B1;
        end
else
begin
        oHD     <= mhd;
        oVD     <= mvd;
        oDEN <= display_area;
        oLCD_R1 <= read_red;
        oLCD_G1 <= read_green;
        oLCD_B1 <= read_blue;
        oLCD_R  <= oLCD_R1;
        oLCD_G  <= oLCD_G1;
        oLCD_B  <= oLCD_B1;
end
end
always@(posedge iCLK or negedge iRST_n)
begin
if (!iRST_n)
begin
```

```
coord_x      <= 0;
coord_y      <= 0;
end
else
begin
  if (display_area)
begin
coord_x      <= x_cnt - (Hsync_Blank-1'h1);
coord_y      <= y_cnt - (Vertical_Back_Porch-1'h1);
end
end
        end
endmodule
```